Homogeneity Analysis

Jan de Leeuw

Version 1.02, November 12, 2015

Contents

Note

The directory http://deleeuwpdx.net/pubfolders/homals has a pdf copy of this article, the complete Rmd file with all code chunks, and R and C files with the code.

Loss Function

In the multivariate analysis techniques presented in this paper the data are measurements or categorizations of n *objects* by p *variables*. Variables are partitioned into m *sets of variables*, with set j containing p_j variables. Thus $\sum_{j=1}^m p_j = p$. The number of variables in a set is also called the *rank* of the set.

Homogeneity Analysis, as defined in this paper, is the minimization of the loss function

$$\sigma(X, H, A) = \frac{1}{m}\sum_{j=1}^m \mathbf{SSQ}\ (X - H_j A_j), \tag{1}$$

where $\mathbf{SSQ}()$ is the sum of squares.

The loss function depends on three sets of parameters X, H and A. One of them, the matrix X, is common to all sets. The data are in the matrix H_j, which codes observations on the p_j variables in set j. Homogeneity analysis does not operate on the variables directly, but on *admissible transformations* of the variables. Matrix A_j has coefficients used to make linear combinations of the variables in set j.

1. The $n \times r$ matrix X of *object scores* is required to be column-centered and normalized by $X'X = I$. The *dimensionality* r is chosen by the user.
2. The $n \times p_j$ matrices of *transformed variables* H_j have columns $h_{j\ell}$, with $\ell = 1, \cdots, p_j$. We require $h_{j\ell} \in \mathcal{K}_{|\ell} \cap \mathcal{S}$, where $\mathcal{K}_{j\ell}$ is a *cone of admissible transformations* and $\mathcal{S}$ is the unit sphere in $\mathbb{R}^n$. For example, $\mathcal{K}_{j\ell}$ can be the cone of all centered vectors in $\mathbb{R}^n$ that are monotone with the data.
3. The $p_j \times r$ matrices A_j of *component loadings* are unconstrained.

Thus we make linear combinations of the transformed variables in such a way that the *set scores* $H_j A_j$ are as close as possible to the object scores X, and consequently as close as possible to each other. This explains the name of the technique: we want to make the m set scores as *homogeneous* as possible. Note that homogeneity analysis is both linear and non-linear, in the sense that it makes linear combinations of non-linear transformations of the variables.

The history of loss function (1) is complicated. Least squares and eigenvalue methods for quantifying multivariate qualitative data were introduced by Guttman (1941). They were taken up by, among others, De Leeuw (1968) and by Benzécri and his students (Cordier 1965). In this earlier work the emphasis was often on optimizing quadratic forms, or ratios of quadratic forms, and not so much on least squares, distance geometry, and so-called *biplots* (Gower and Hand 1996).

In De Leeuw (1974) a first attempt was made to unify most classical descriptive multivariate techniques using a single least squares loss function and a corresponding *alternating least squares (ALS)* optimization method. Guttman's quantification method, which later became known as *multiple correspondence analysis*, was merged with linear and nonlinear principal component analysis in the HOMALS/PRINCALS techniques and programs (De Leeuw and Van Rijckevorsel 1980). The homogeneity analysis loss function that was

chosen ultimately, for example in the work of Van der Burg, De Leeuw, and Verdegaal (1988), had been used earlier by Carroll (1968) in multi-set canonical correlation analysis.

In the Gifi system (Gifi 1990, Michailidis and De Leeuw (1998)) a slightly different parametrization, and a correspondingly different ALS algorithm, were used. The loss function used by Gifi is

$$\sigma(X,Y)=\frac{1}{m}\sum_{j=1}^m\mathbf{SSQ}\ (X-\sum_{\ell=1}^{p_j}G_{j\ell}Y_{j\ell}),\tag{2}$$

where the $G_{j\ell}$ are known spanning matrices for the cones of admissible transformations, and the $Y_{j\ell}$ are $k_{j\ell}\times p$ matrices of *category quantifications*. By using the full rank decompositions $Y_{j\ell}=Z_{j\ell}A_{j\ell}$ we can show that (1) and (2) are essentially the same. We feel that the new parametrization in terms of H_j and and A_j has some conceptual and computational advantages.

Algorithm

The standard way to minimize loss function (2) is implemented in the `OVERALS` program (Van der Burg, De Leeuw, and Verdegaal 1988, Meulman and Heiser (2011)). It is also the one used in the `homals` package (De Leeuw and Mair 2009).

In this paper the algorithm is different because we use the loss function (1). We still use ALS, which means in this case that we cycle through three substeps in each iteration. We update A for given X and H, we then update X for given H and A, and finally we update H for given X and A. Algorithm A goes as follows.

0. Set $k=0$ and start with some $X^{(0)},H^{(0)},A^{(0)}$.
1. $X^{(k+1)}=\mathbf{ortho}(\mathbf{center}(\sum_{j=1}^m H_j^{(k)}A_j^{(k)}))$.
2. For $j=1,\cdots,m$ compute $A_j^{(k+1)}=\{H_j^{(k)}\}^+X^{(k+1)}$.
3. For $j=1,\cdots,m$ and $s=1,\cdots p_j$ compute $h_{js}^{(k+1)}=\mathbf{proj}_{\mathcal{K}_{js}\cap\mathcal{S}}((X^{(k+1)}-\sum_{t<s}h_{jt}^{(k+1)}\{a_{jt}^{(k+1)}\}'-\sum_{t>s}h_{jt}^{(k)}\{a_{jt}^{(k+1)}\}')a_s^{(k+1)})$.
4. If converged stop. Else $k\leftarrow k+1$ and go to step 1.

In step 1 we use superscript + for the Moore-Penrose inverse. In step 2 the center operator does column centering, the ortho operator finds an orthonormal basis for the column space of its argument.

The complicated part is step 4, the *optimal scaling*, i.e. the updating of H_j for given X and A_j. We cycle through the variables in the set, each time projecting a single column on the cone of admissible transformations of the variable, and then normalizing the projection to length one. The *target*, i.e. the vector we are projecting, is complicated, because the other variables in the same set must be taken into account.

In order to simplify the optimal scaling computations within an iteration we can use majorization (De Leeuw 1994, Heiser (1995), Lange, Hunter, and Yang (2000), J. De Leeuw (2015a)). This has the additional benefit that the optimal scaling step becomes embarassingly parallel. We expand the loss for set j around a previous solution $\tilde H_j$.

$$\mathbf{SSQ}(X-H_jA_j)=\mathbf{SSQ}(X-\tilde H_jA_j)-2\mathbf{tr}\ (H_j-\tilde H_j)'(X-\tilde H_jA_j)A_j'+\mathbf{tr}\ A_j'(H_j-\tilde H_j)'(H_j-\tilde H_j)A_j.$$

Now

$$\mathbf{tr}\ (H_j-\tilde H_j)A_jA_j'(H_j-\tilde H_j)'\leq\kappa_j\ \mathbf{tr}\ (H_j-\tilde H_j)'(H_j-\tilde H_j),$$

where κ_j is the largest eigenvalue of $A_j'A_j$. Thus

$$\mathbf{SSQ}(X-H_jA_j)\leq\mathbf{SSQ}(X-\tilde H_jA_j)+\kappa_j\ \mathbf{SSQ}(H_j-U_j)-\frac{1}{\kappa_j}\ \mathbf{SSQ}((X-\tilde H_jA_j)A_j'),$$

where U_j is the *target*

$$U_j=\tilde H_j+\frac{1}{\kappa_j}(X-\tilde H_jA_j)A_j'.\tag{3}$$

It follows we can update the optimal scaling of the variables by projecting the columns of U_j on their respective cones and then normalizing. See De Leeuw (1975) for results on normalized cone regression. This can be done for all variables in the set separately, without taking any of the other variables in the set (or in any of the other sets) into account. Thus the optimal scaling is easy to parallellize. The resulting algorithm B is as follows.

0. Set $k = 0$ and start with some $X^{(0)}, H^{(0)}, A^{(0)}$.
1. $X^{(k+1)} = \textbf{ortho}(\textbf{center}(\sum_{j=1}^m H_j^{(k)} A_j^{(k)}))$.
2. For $j = 1, \cdots, m$ compute $A_j^{(k+1)} = \{H_j^{(k)}\}^+ X^{(k+1)}$.
3. For $j = 1, \cdots, m$ compute $U_j^{(k+1)} = H_j^{(k)} + \frac{1}{\kappa_j}(X^{(k+1)} - H_j^{(k)} A_j^{(k+1)})\{A_j^{(k+1)}\}'$ and for $s = 1, \cdots p_j$ compute $h_{js}^{(k+1)} = \textbf{proj}_{\mathcal{K}_{js} \cap \mathcal{S}}(u_{js}^{(k+1)})$.
4. If converged stop. Else $k \leftarrow k + 1$ and go to step 1.

Implementation Details

If we follow the ALS strategy strictly the **ortho**() operator should be implemented using Procrustus rotation (Gibson 1962). Thus if $Z = K\Lambda L'$ is the singular value decomposition of X, then $\textbf{ortho}(Z) = KL'$. Note, however, that any other basis for the column space of Z merely differs from the Procrustus basis by a rotation. And this rotation matrix will carry unmodified into the upgrade of A_j in step 2 of the algorithm, and thus after steps 1 and 2 the loss will be the same, no matter whch rotation we select. In our algorithm we use the QR decomposition to find the basis, using the Gram-Schmidt code from J De Leeuw (2015).

We implement the cone restrictions by the constraints $h_{js} = G_{js} z_s$ in combination with $T_{js} h_{js} \geq 0$. Thus the transformed variables must be in the intersection of the subspace spanned by the columns of the *transformation basis* G_{js} and the polyhedral convex cones of all vectors h such that $T_{js} h \geq 0$. We suppose that all columns of the G_{js} add up to zero, and we require, in addition, the normalization $SSQ(h_{js}) = 1$.

In earlier homogeneity analysis work, summarized for example in Gifi (1990) or Michailidis and De Leeuw (1998), the transformation basis matrices G_{js} were binary zero-one matrices, indicating category membership. The same is true for the software in IBM SPSS Categories (Meulman and Heiser 2011) or in the `R` package `homals` (De Leeuw and Mair 2009). In this paper we extend the current homogeneity analysis software using *B-spline bases*, which provide a form of fuzzy non-binary coding suitable for both categorical and numerical variables (Van Rijckevorsel and De Leeuw 1988). These generalizations were already discussed in De Leeuw, Van Rijckevorsel, and Van der Wouden (1981) and Gifi (1990), but corresponding easily accessible software was never released.

We use the code described in J. De Leeuw (2015b) to generate B-spline bases. Note that for coding purposes binary indicators are B-splines of degree zero, while polynomials are B-splines without interior knots. Also note that binary indicators can be created for qualitative non-numerical variables, for which B-splines are not defined. We have added the option using degree -1 to bypass the B-spline code and generate an indicator matrix. Throughout we first orthonormalize the basis matrices G_{js}, using the Gram-Schmidt code from J De Leeuw (2015).

The matrices T_{js} in the homogeneous linear inequality restrictions that define the cones $\mathcal{K}_{js}$ can be used to define monotonicity or convexity of the resulting transformations. In the current implementation we merely allow for monotonicity, which means the T_{js} do not have to be stored. The transformations for each variable can be restricted to be increasing, or they can be unrestricted. By using splines without interior knots we allow in addition for polynomial transformations, which again can be restricted to be either monotonic or not. This covers the previous Gifi types *nominal*, *ordinal*, and *numerical*, which were of course designed for categorical variables with a small number of categories. Note that it is somewhat misleading to say we are fitting monotone splines or polynomials, we are mainly requiring monotonicity at the data points.

Missing data are incorporated in the definition of the cones of transformations by using a G_{js} which is the direct sum of a spline basis for the non-missing and an identity matrix for the missing data. This is called *missing data multiple* in Gifi (1990). There are no linear inequality restrictions on the quantifications of the missing data.

Associated Eigenvalue Problems

Associated with the problem of minimizing loss function (1) are some eigen and singular value problems defined by the matrices H_j. This has been discussed extensively in Gifi (1990), and there are some more recent discussions in Tenenhaus and Tenenhaus (2011) and Van der Velden and Takane (2012).

Suppose H_j^+ is the Moore-Penrose inverse of H_j and $P_j = H_j H_j^+$ is the orthogonal projector associated with the column space of H_j. Then

$$\min_{A_j} \sigma(X, H, A) = \frac{1}{m} \sum_{j=1}^{m} (r - \mathbf{tr}\ X' P_j X) = r - \mathbf{tr}\ X' P_\star X,$$

with $P_\star$ the average of the projectors P_j. Thus

$$\min_{X'X=I} \min_{A_j} \sigma(X, H, A) = r - \sum_{s=1}^{r} \lambda_s(P_\star),$$

where λ_s are the ordered eigenvalues of $P_\star$ (from large to small). Thus we see that homogeneity analysis chooses the H_j, i.e. transforms or quantifies the variables, in such a way that the sum of the r largest eigenvalues of $P_\star$ is maximized.

Now consider alternative restrictions where we do not normalize X, but we normalize the loadings A by requiring that

$$\frac{1}{m} \sum_{j=1}^{m} A_j' D_j A_j = I,$$

where $D_j = H_j' H_j$. Also define $C_{jv} = H_j' H_v$. We can collect the matrices C_{jv} in an $p \times p$ super-matrix C, which we we will call the *Burt matrix*. Note that if $H = \begin{bmatrix} H_1 & H_2 & \cdots & H_m \end{bmatrix}$, then $C = H'H$. The matrix D is defined as the direct sum of the D_j, i.e. it consists of the diagonal submatrices of C.

The minimum of loss over unrestricted X for fixed A and H is attained at the average of the set scores $X = \frac{1}{m} \sum_{j=1}^{m} H_j A_j$, and thus

$$\min_X \sigma(X, H, A) = r - \frac{1}{m^2} \mathbf{tr}\ A'CA,$$

and

$$\min A'DA = mI \min_X \sigma(X, H, A) = r - \sum_{s=1}^{r} \lambda_s(C, mD),$$

where the λ_s are now the generalized eigenvalues of the pair C and mD.

If we define K by

$$K = m^{-\frac{1}{2}} \begin{bmatrix} H_1 (H_1' H_1)^{-\frac{1}{2}} & \cdots & H_m (H_m' H_m)^{-\frac{1}{2}} \end{bmatrix},$$

with $(H_j' H_j)^{-\frac{1}{2}}$ the Moore-Penrose inverse of the symmetric square root. Then $P_\star = KK'$ and the non-zero eigenvalues of $P_\star$ are the same as those of $K'K$, which in turn are equal to the generalized eigenvalues of the pair (C, mD). Thus homogeneity analysis can also be interpreted as transforming the variables in such a way that the sum of the p largest generalized eigenvalues of (C, mD) is maximized.

Special Cases

1. If there are only two sets the generalized eigenvalue problem for the Burt matrix becomes

$$\begin{bmatrix} D_1 & C_{12} \\ C_{21} & D_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = 2\lambda \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix},$$

   which we can rewrite as

$$\begin{aligned} C_{12} a_2 &= (2\lambda - 1) D_1 a_1, \\ C_{21} a_1 &= (2\lambda - 1) D_2 a_2, \end{aligned}$$

from which we see that homogeneity analysis maximizes the sum of the r largest canonical correlations between H_1 and H_2. See also Van der Velden (2012).

2. Suppose all m sets each contain only a single variable. Then the Burt matrix is the correlation matrix of the H_j, which are all $n \times 1$ matrices in this case. It follows that homogeneity analysis maximizes the sum of the r largest eigenvalues of the correlation matrix over transformations, i.e. homogeneity analysis is *nonlinear principal component analysis* (De Leeuw 2014).

3. Suppose all basis matrices $G_{j\ell}$ in set j are the same, say equal to G_j. Then the set scores $H_j A_j$ are equal to $G_j Z_j A_j$, which we can write simply as $G_j Y_j$. Thus loss must be minimized over X and the Y_j. If all G_j are binary indicators of categorical variables, and the m sets are all of rank one, then homogeneity analysis is *multiple correspondence analysis* (MCA). The set scores $G_j Y_j$ are k_j different points, with k_j the number of categories of the variable, usually much less than n. The plot connecting the set scores to the object scores is called the *star plot* of the variable.

4. More generally, we can include an arbitrary number of *copies* of a variable in a set by using the same basis matrix G_j a number of times. As soon as we have decided how many copies to include, the algorithm can forget all about the fact that some variables are copies and just treat them like any other variable. The notion of copies replaces the notion of the *rank of a quantification* used in traditional Gifi, which in turn generalizes the distinction between *single* and *multiple* quantifications.

5. If the second set only contains a single copy of a single variable then we choose transformations that maximize the multiple correlation of that variable and the variables in the first set.

6. If the second set contains more than one copy of a single variable and we use binary indicator coding for that variable, then we optimize the eigenvalue (between/within ratio) sums for a canonical discriminant analysis.

MCA Example: Thirteen Personality Scales

Our first example is a small data set from the `psych` package (Revelle 2015) of five scales from the Eysenck Personality Inventory, five from a Big Five inventory, a Beck Depression Inventory, and State and Trait Anxiety measures.

```
data(epi.bfi, package = "psych")
epi <- epi.bfi
epi_knots <- lapply (epi, function (x) fivenum (x)[2:4])
epi_degrees <- rep (0, 13)
epi_ordinal <- rep (FALSE, 13)
epi_copies <- rep (2,13)
epi_sets <- 1:13
```

We perform a two-dimensional MCA, using degree zero and inner knots at the three quartiles for all 13 variables.

```
h <- homals(epi, epi_knots, epi_degrees, epi_ordinal, epi_sets, epi_copies, verbose = FALSE)
```

We have convergence in 260 iterations to loss 0.7478043. The object scores are in figure 8.

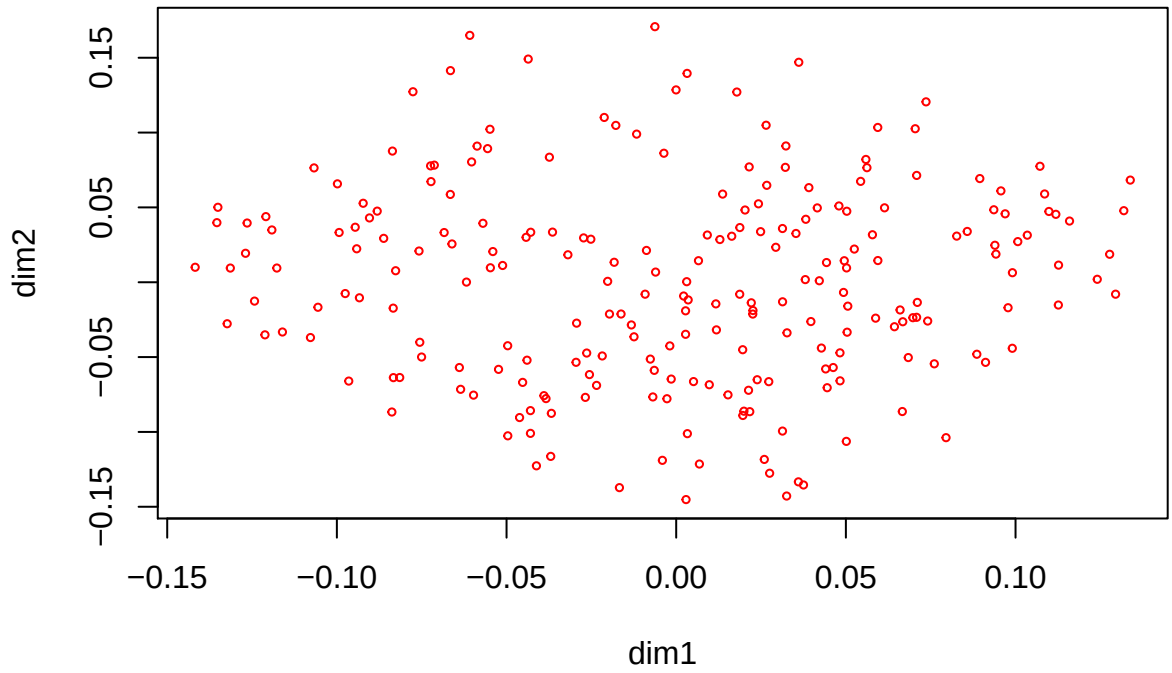
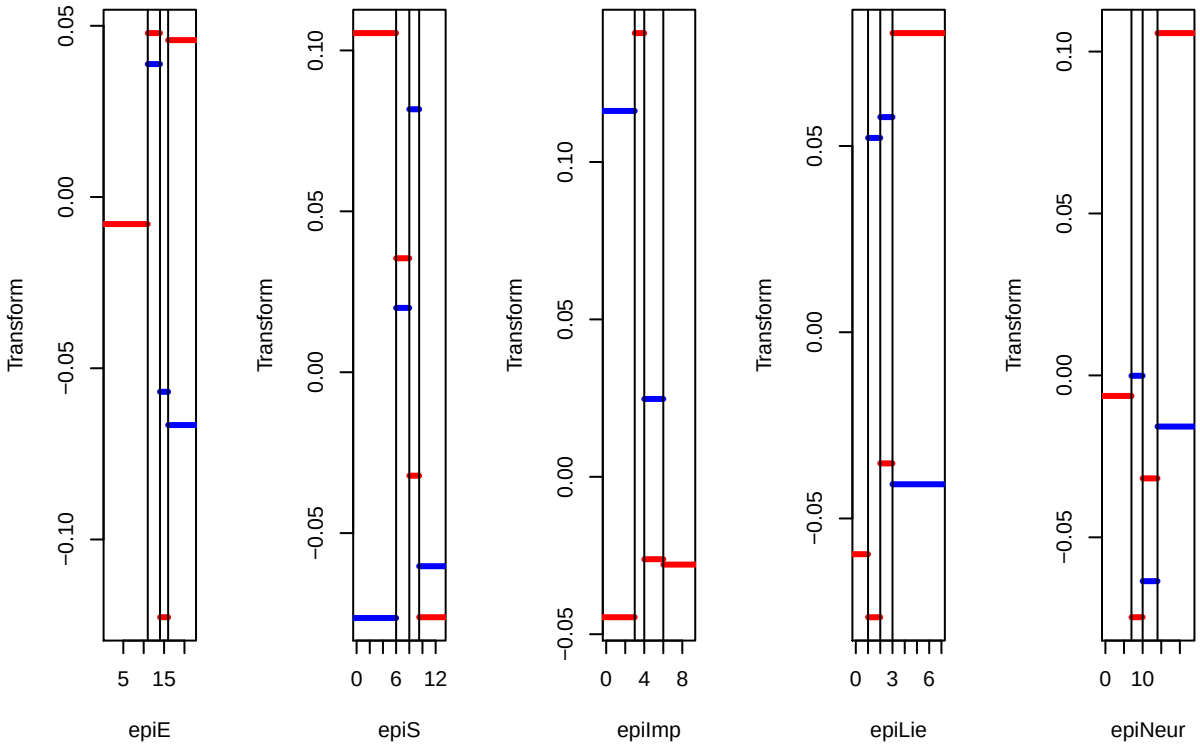


Figure 8: Personality Scales, Object Scores, Multiple Nominal, Degree Zero

Figure 9 has the $G_j Y_j$ for each of the thirteen variables, with the first dimension in red, and the second dimension in blue. Because the degree of the splines is zero, these *transformation plots* show step functions, with the steps at the knots, which are represented by vertical lines.



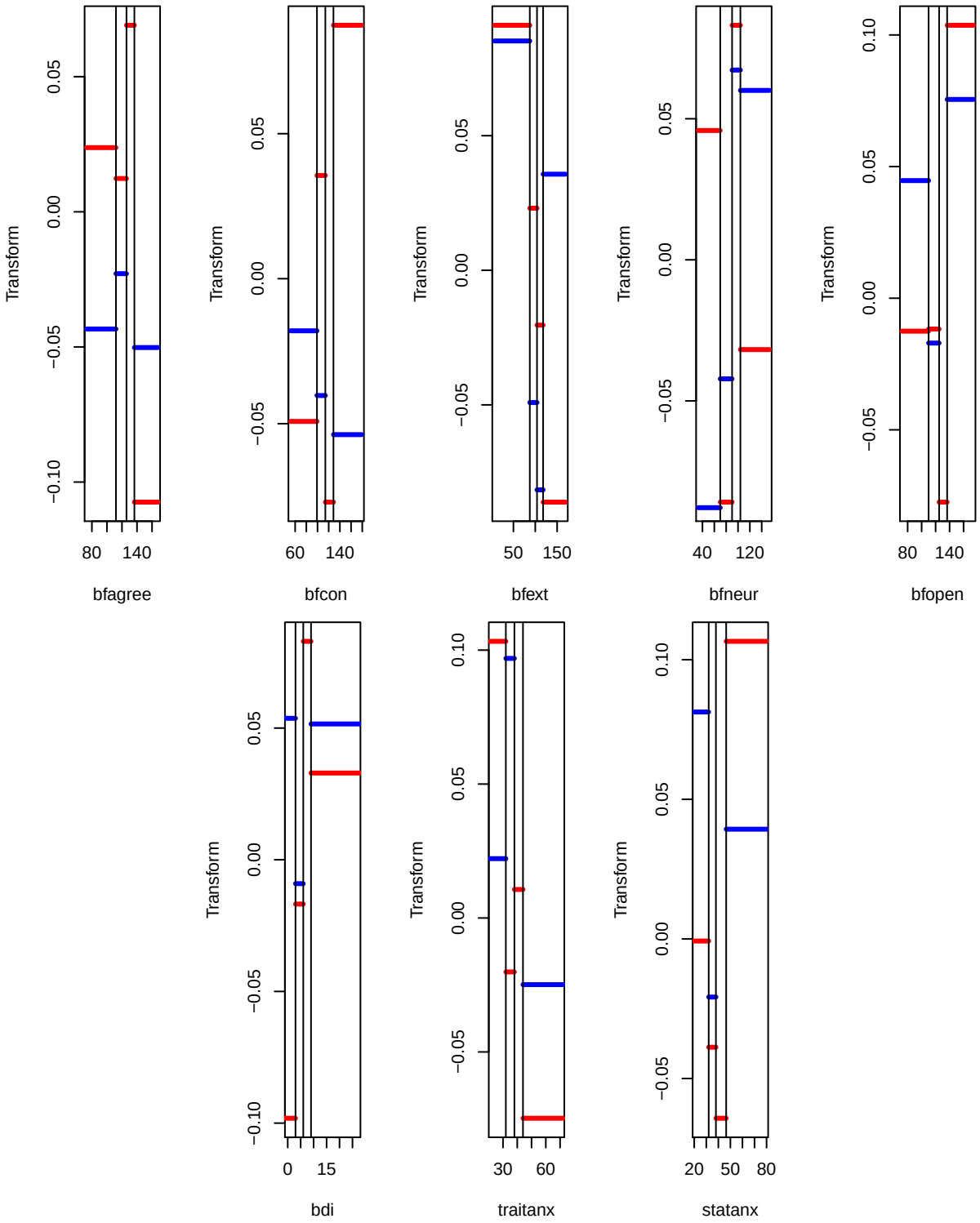
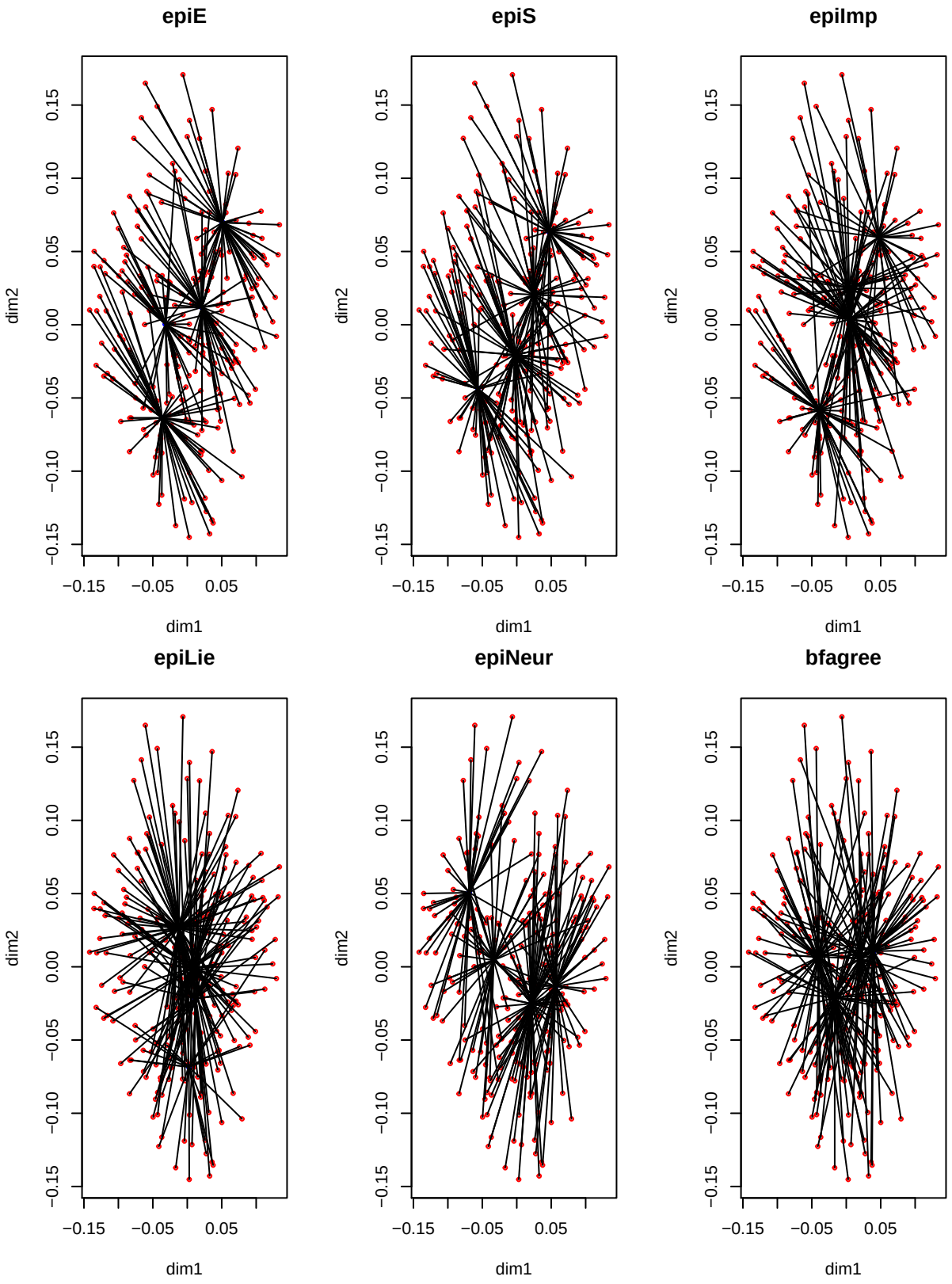
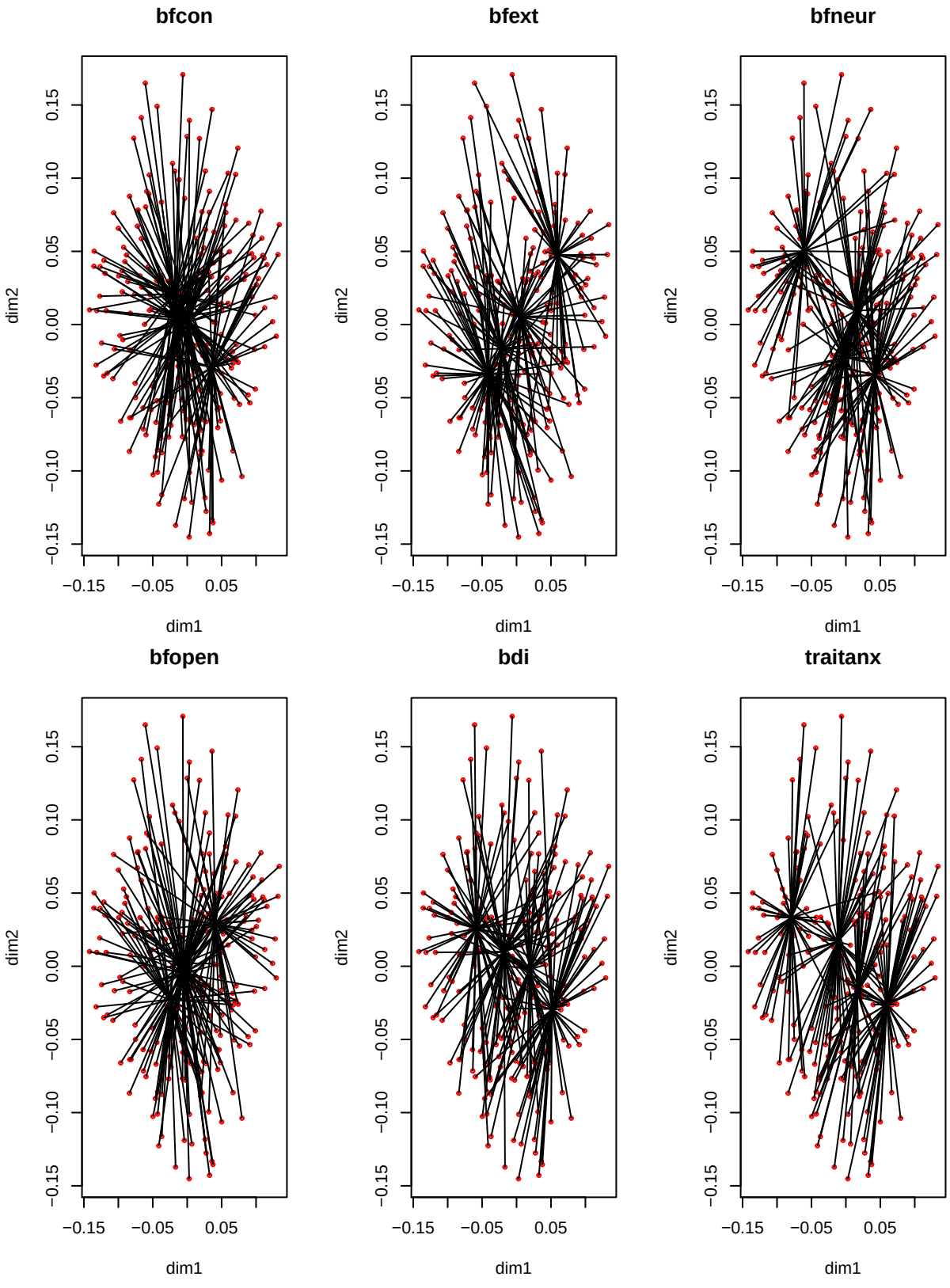


Figure 9: Personality Scales, Transformations, Multiple Nominal, Degree Zero

The thirteen star plots are in figure 9.


epiE
epiS
epiImp
epiLie
epiNeur
bfagree
dim1
dim2
−0.15
−0.05
0.05
0.00
0.10
0.15
−0.10


bfcon
bfext
bfneur
bfopen
bdi
traitanx
dim1
dim2
−0.15
−0.05
0.05
0.00
0.10
0.15

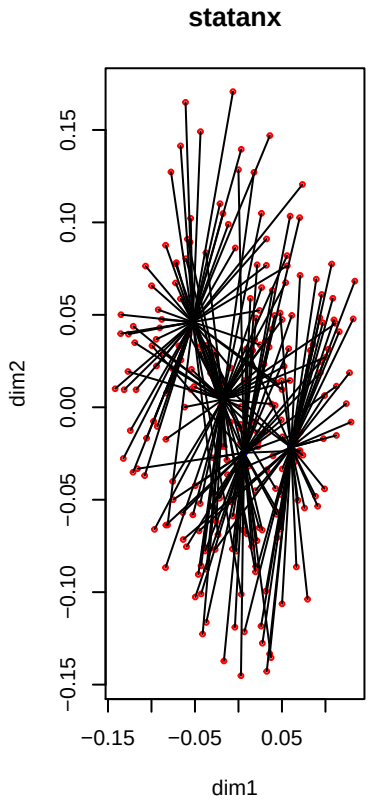


Figure 10: Personality Scales, Star Plots, Multiple Nominal, Degree Zero

Now change the degree to two for all variables, i.e. fit piecewise quadratic polynomials which are differentiable at the knots. We still have two copies for each variable, and these two copies define the sets.

```
epi_degrees <- rep (2, 13)
h <- homals(epi, epi_knots, epi_degrees, epi_ordinal, epi_sets, epi_copies, verbose = FALSE)
```

We have convergence in 785 iterations to loss 0.7179135. The object scores are in figure 11 and the transformation plots in figure 12.

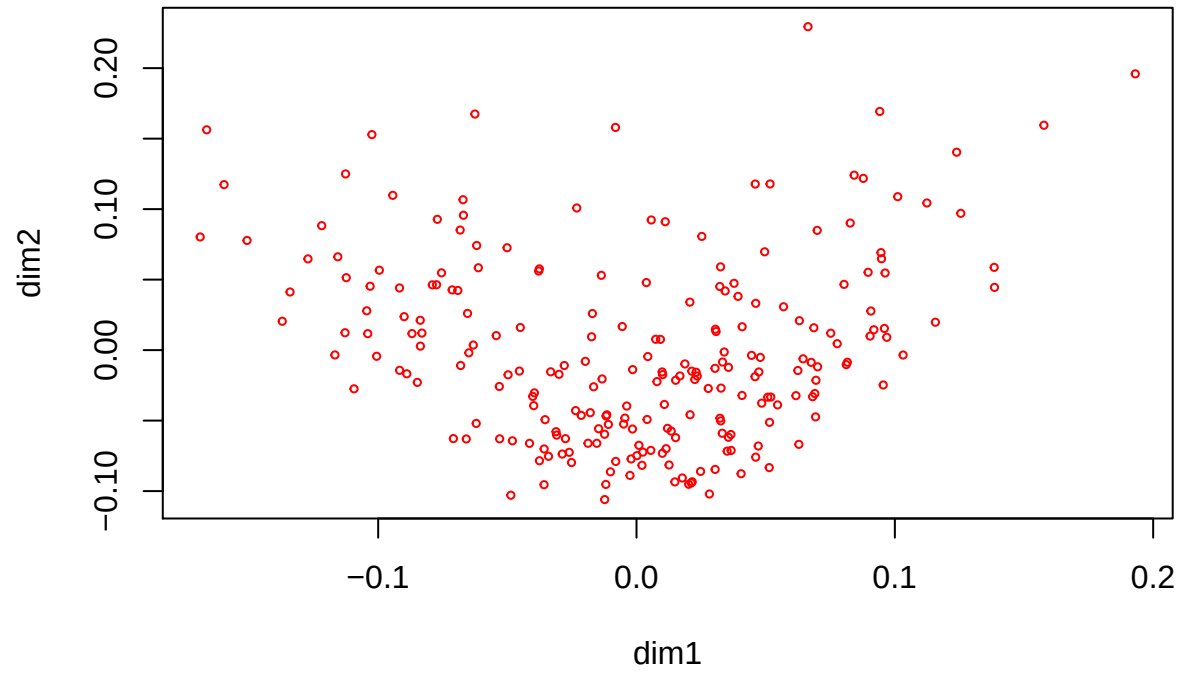
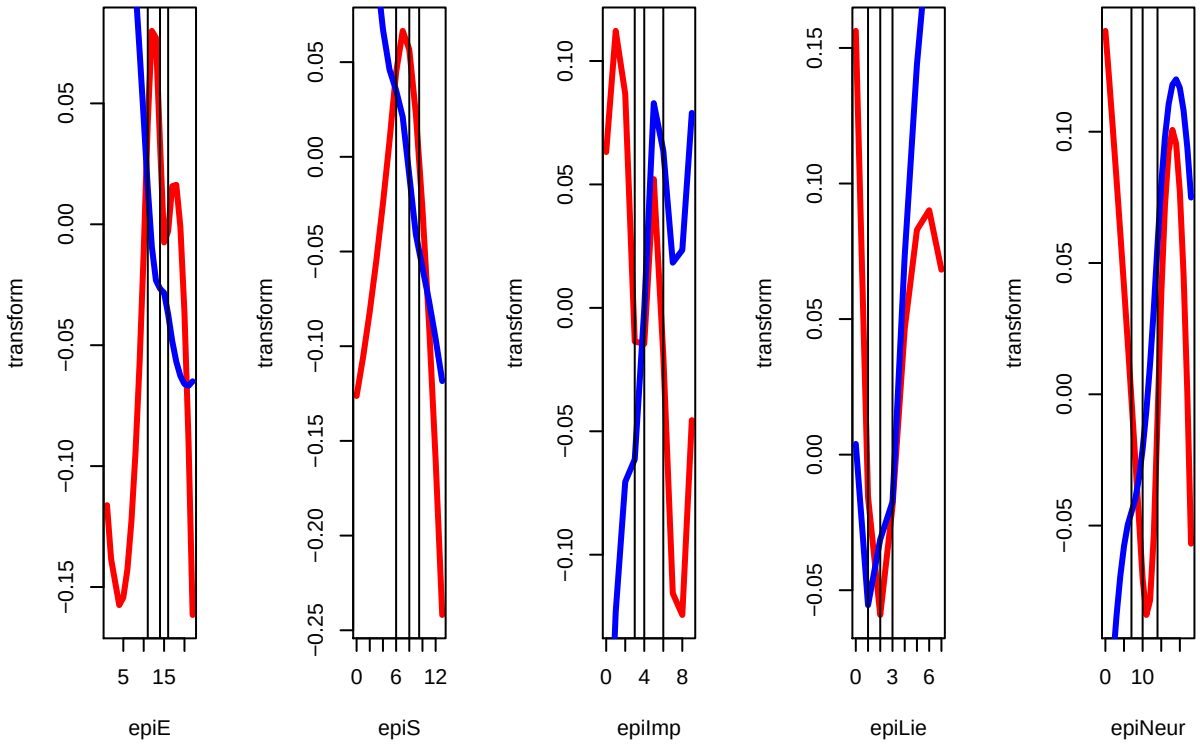


Figure 11: Personality Scales, Object Scores, Multiple Nominal, Degree Two



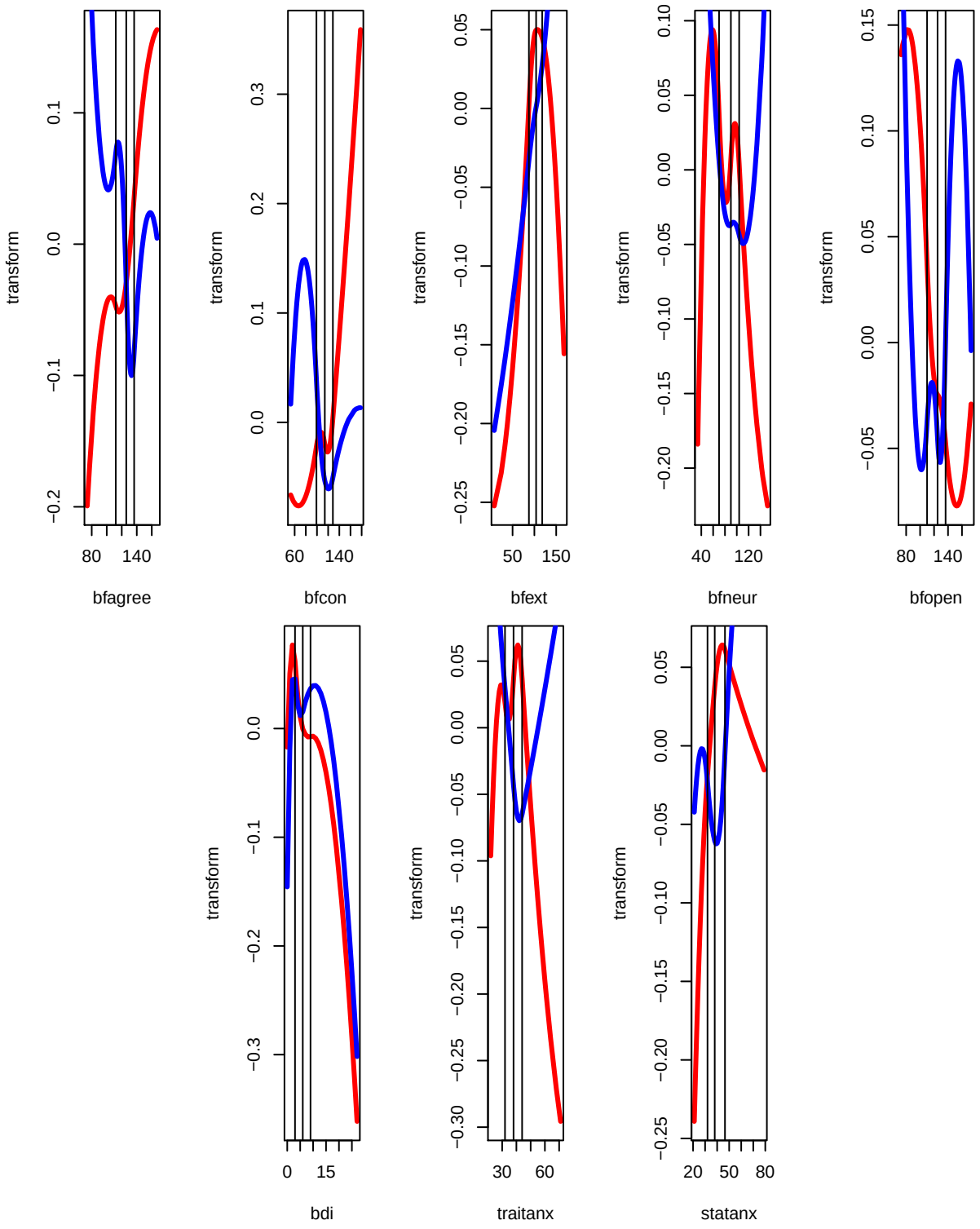


Figure 12: Personality Scales, Transformations, Multiple Nominal, Degree Two

NLPCA Example: Thirteen Personality Scales

We use the same data as before for an NLPCA with all sets of rank one, all variables ordinal, and splines of degree 2.

```
library(nnls)
epi_copies <- rep (1, 13)
epi_ordinal <- rep (TRUE, 13)
h <- homals(epi, epi_knots, epi_degrees, epi_ordinal, epi_sets, epi_copies, verbose = FALSE)
```

In 19 iterations we find minimum loss 0.7330982. The object scores are in figure 13 and the transformation plots in figure 14. NLPCA maximizes the sum of the two largest eigenvalues of the correlation matrix of the variables. Before transformation the eigenvalues are 4.0043587, 2.6702003, 1.9970912, 0.8813983, 0.6571463, 0.6299946, 0.5246896, 0.4657022, 0.3457515, 0.3403361, 0.2767531, 0.1835449, 0.0230333, after transformation they are 4.1939722, 2.7454868, 1.604906, 0.8209072, 0.7184825, 0.677183, 0.51865, 0.4545214, 0.4200148, 0.351787, 0.2928574, 0.1699557, 0.0312759. The sum of the first two goes from 6.674559 to 6.9394591.

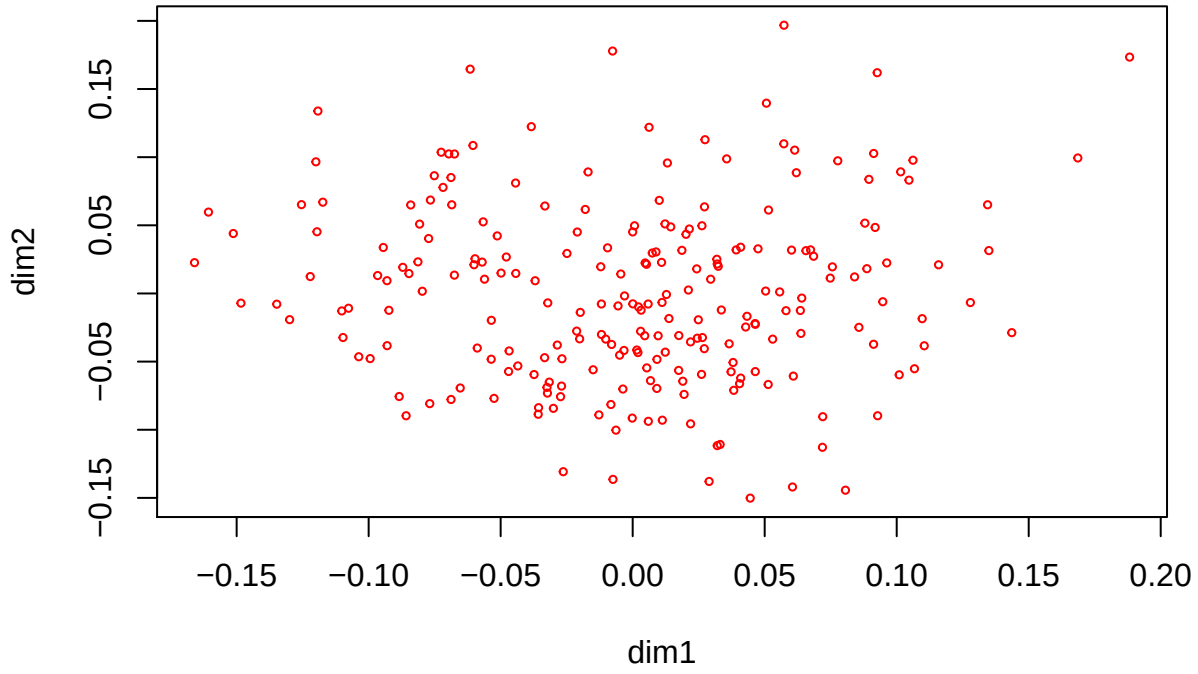
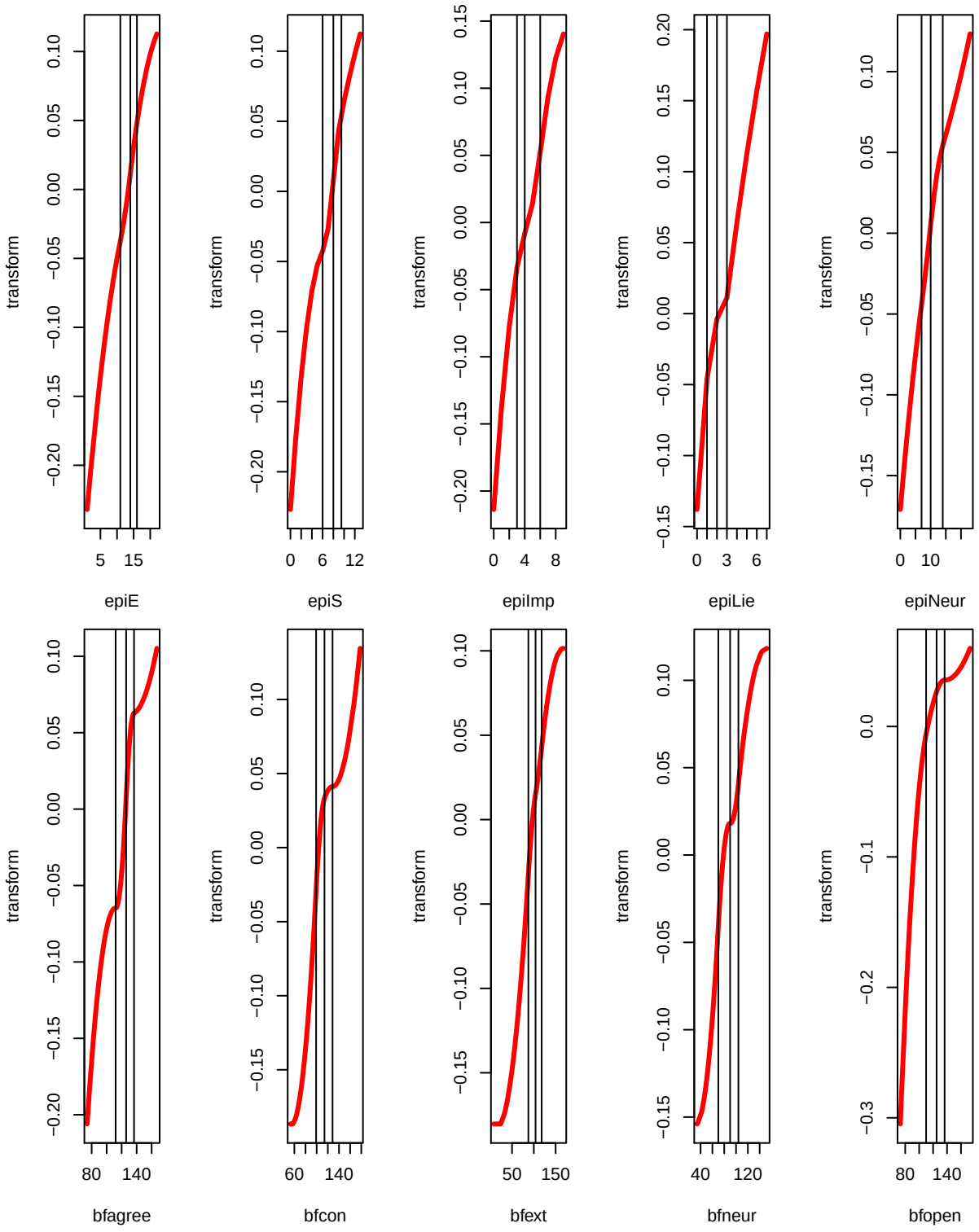


Figure 13: Personality Scales, Object Scores, Single Ordinal, Degree Two


transform
epiE
epiS
epiImp
epiLie
epiNeur
bfagree
bfcon
bfext
bfneur
bfopen

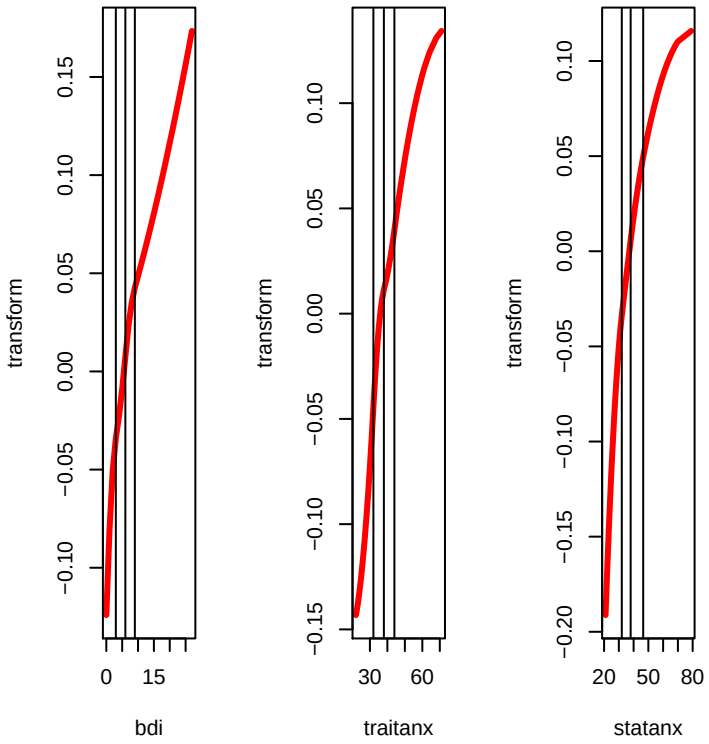


Figure 14: Personality Scales, Transformations, Single Ordinal, Degree Two

We repeat the analysis with ordinal variables of degree two, without interior knots. Thus we the transformation plots will be quadratic polynomials that are monotone over the range of the data.

```
epi_knots <- lapply (1:13, function (j) numeric(0))
h <- homals(epi, epi_knots, epi_degrees, epi_ordinal, epi_sets, epi_copies, verbose = FALSE)
```

In 20 iterations we find minimum loss 0.7393666. The object scores are in figure 15 and the transformation plots in figure 16. The eigenvalues are now 4.0828642, 2.6936186, 1.8391342, 0.8732231, 0.6666505, 0.6491709, 0.5390077, 0.459182, 0.3632868, 0.3471175, 0.2845394, 0.1782232, 0.023982, with sum of the first two equal to 6.7764828.

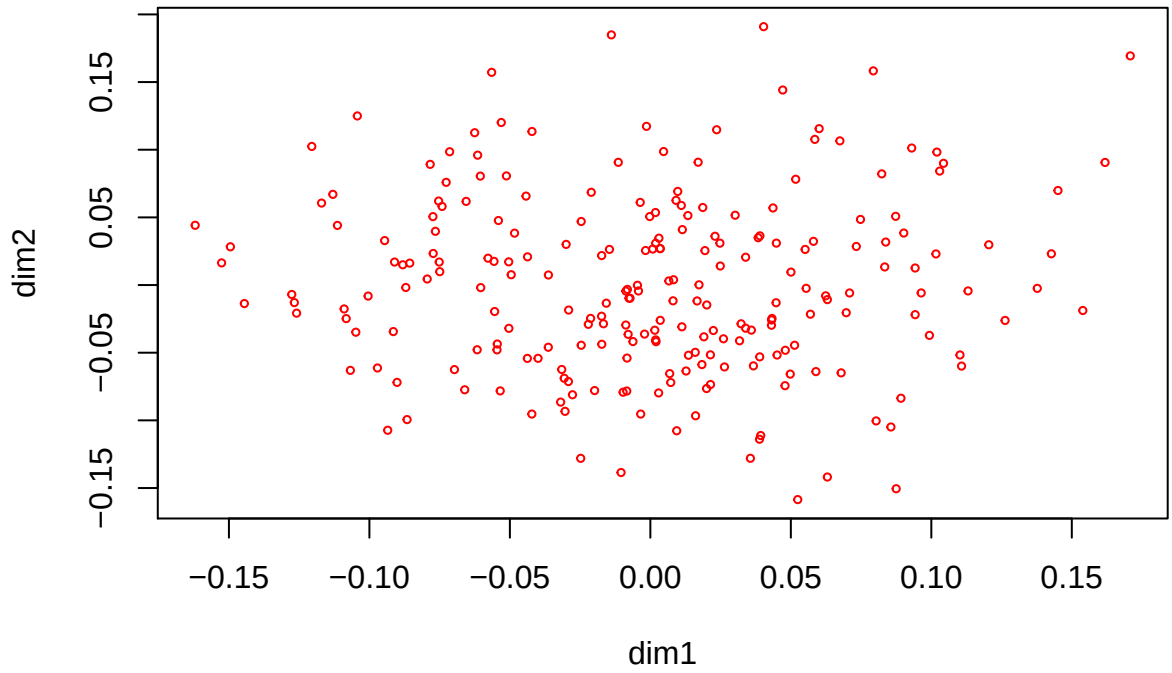
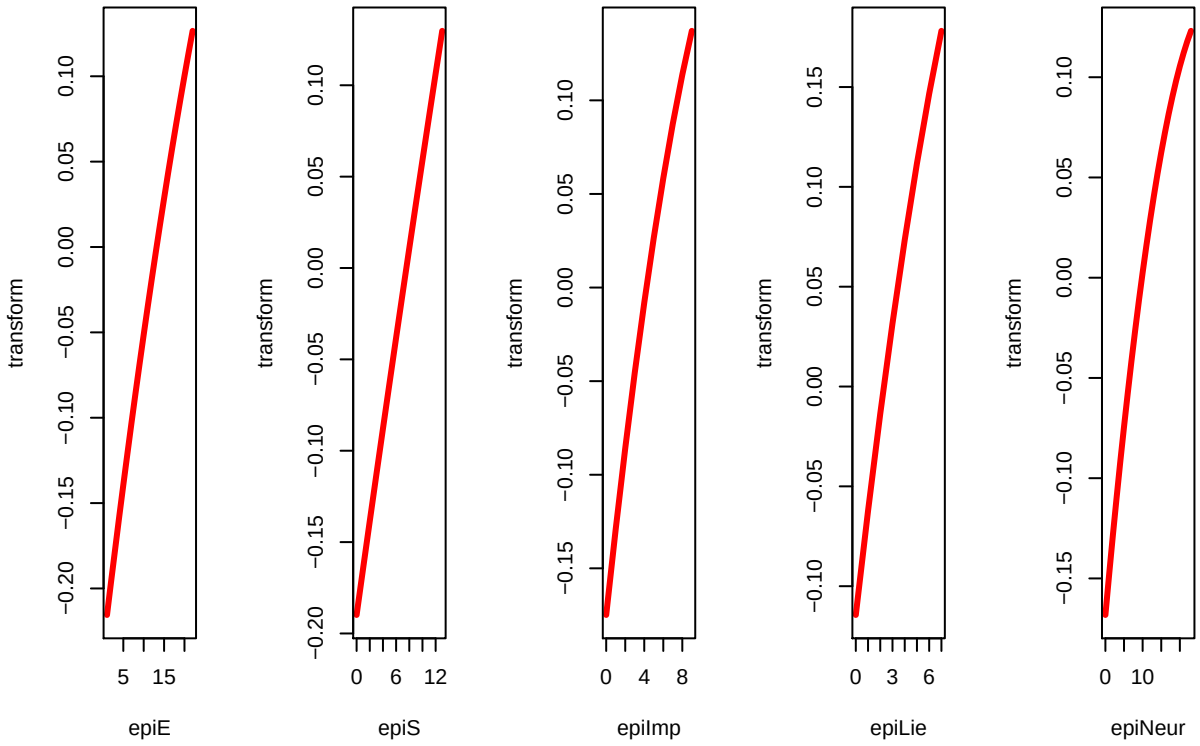


Figure 15: Personality Scales, Object Scores, Single Numerical, Degree Two



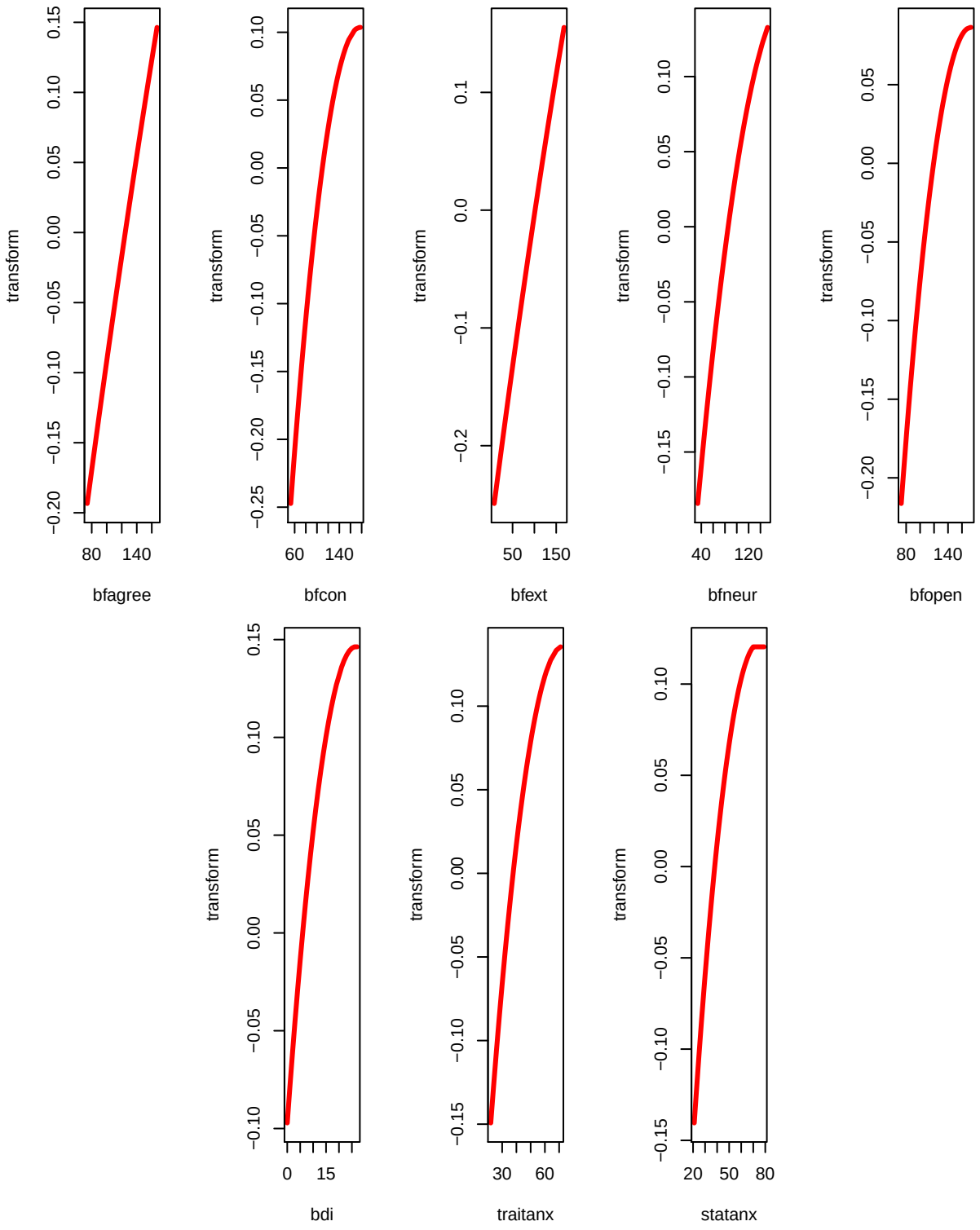


Figure 16: Personality Scales, Transformations, Single Numerical, Degree Two

Regression Example: Gases with Convertible Components

We analyze a regression example, using data from Neumann, previously used by Willard Gibbs, and analyzed with regression in a still quite readable article by Wilson (1926). Wilson's analysis was discussed and modified using splines in Gifi (1990, 370–76). In the regression analysis in this section we use two copies of temperature,

with spline degree zero, and the first copy ordinal. For pressure and the dependent variable density we use a single ordinal copy with spline degree two.

```
data (neumann, package = "homals")
neumann_knots <- lapply (neumann, function (x) fivenum (x)[2:4])
neumann_degrees <- c(0,2,2)
neumann_ordinal <-c(TRUE, TRUE, TRUE)
neumann_copies <- c(2,1,1)
neumann_sets <- c(1,1,2)
```

```
h <- homals(neumann, neumann_knots, neumann_degrees, neumann_ordinal, neumann_sets, neumann_copies, itma
```

In 47 iterations we find minimum loss 0.0268055, corresponding with a multiple correlation of 0.8956526. The object scores are in figure 17 plotted against the original variables (not the transformed variables), and the transformation plots in are figure 18.

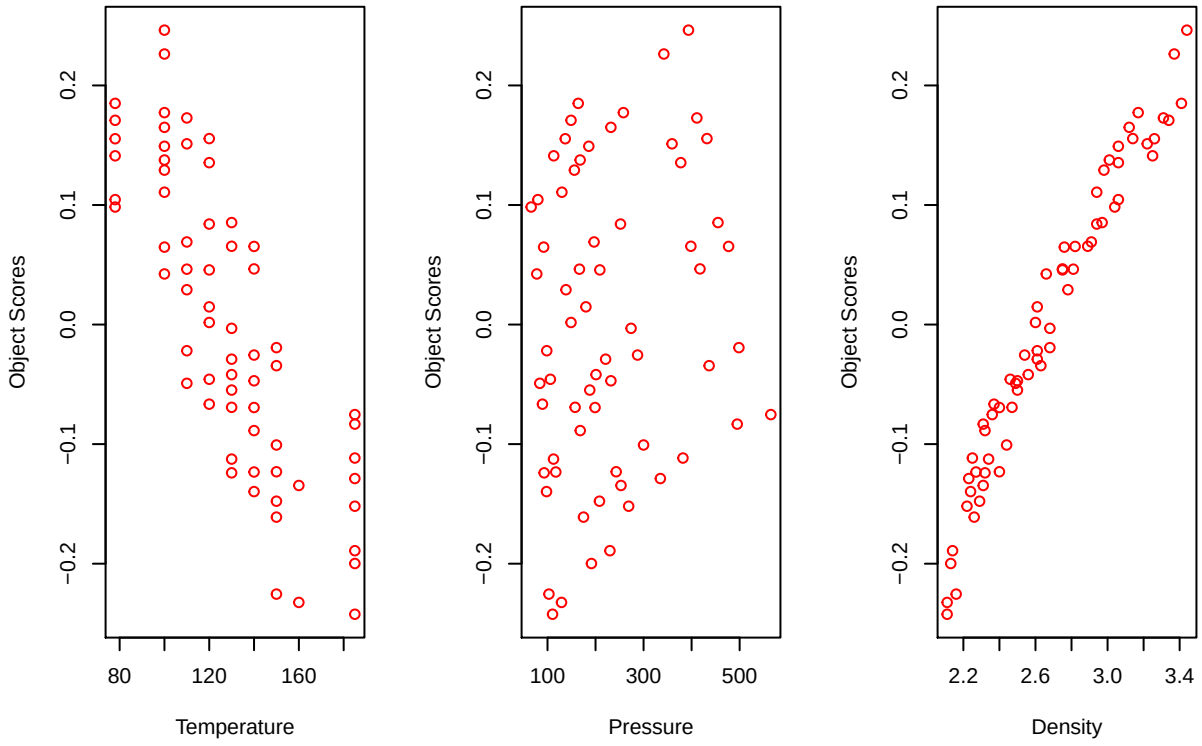


Figure 17: Gases with Convertible Components, Objects Scores

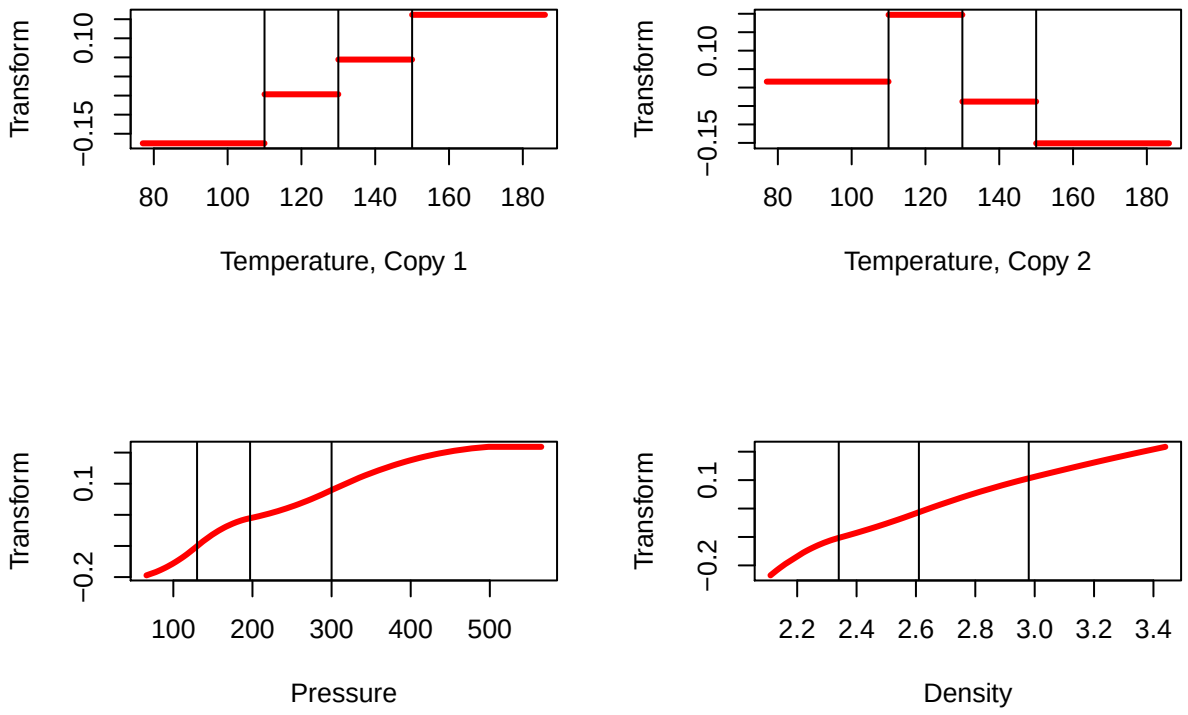


Figure 18: Gases with Convertible Components, Transformations

Discriminant Analysis Example: Iris data

The next example illustrates (canonical) discriminant analysis, using the obligatory Anderson-Fisher iris data. Since there are three species of iris, we use two copies for the species variable. The other four variables are in the same set, they are transformed using piecewise linear monotone splines with five knots.

```
data(iris, package="datasets")
iris_vars <- names(iris)
iris[[5]] <- as.numeric (iris[[5]])
iris_knots <- as.list(1:5)
for (i in 1:4) iris_knots[[i]] <- quantile (iris[[i]], (1:5) / 6)
iris_knots[[5]] <- 1:3
iris_degrees <- c(1,1,1,1,0)
iris_ordinal <- c (TRUE, TRUE, TRUE, TRUE, FALSE)
iris_copies <- c (1,1,1,1,2)
iris_sets <- c(1,1,1,1,2)
```

```
h<-homals(iris, iris_knots,iris_degrees,iris_ordinal,iris_sets,iris_copies, verbose = FALSE)
```

In 126 iterations we find minimum loss 0.0307911. The object scores are in figure 19 plotted against the original variables (not the transformed variables), and the transformation plots are in figure 20.

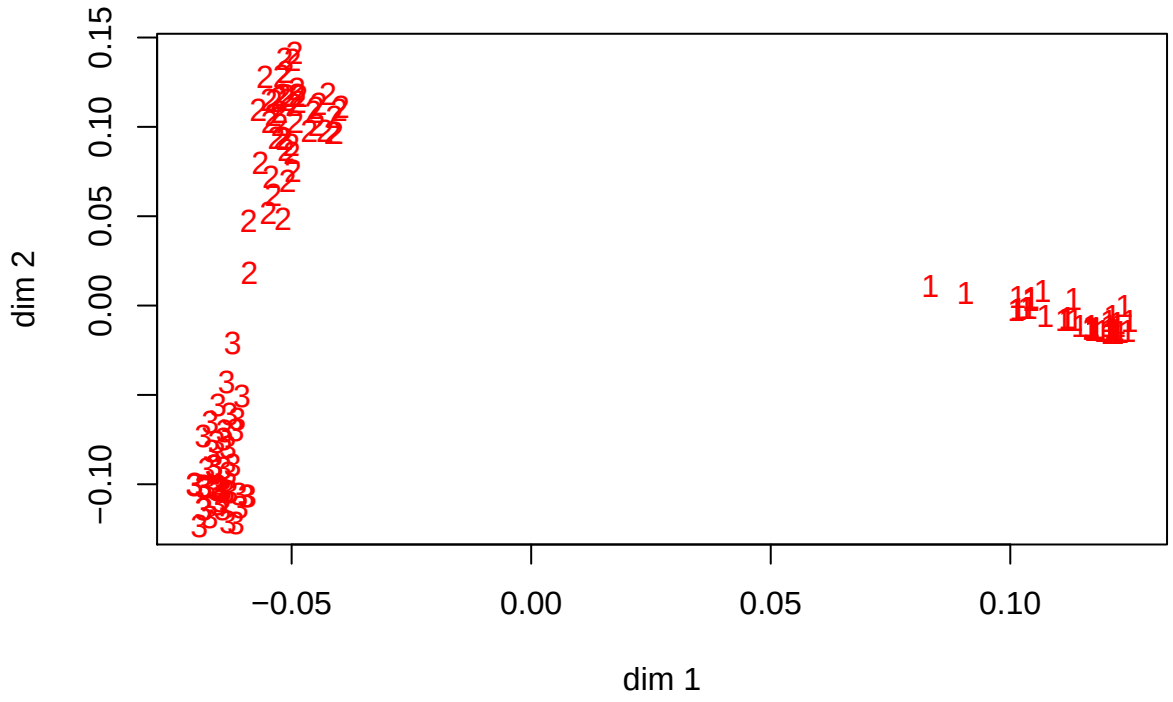


Figure 19: Iris Data, Objects Scores

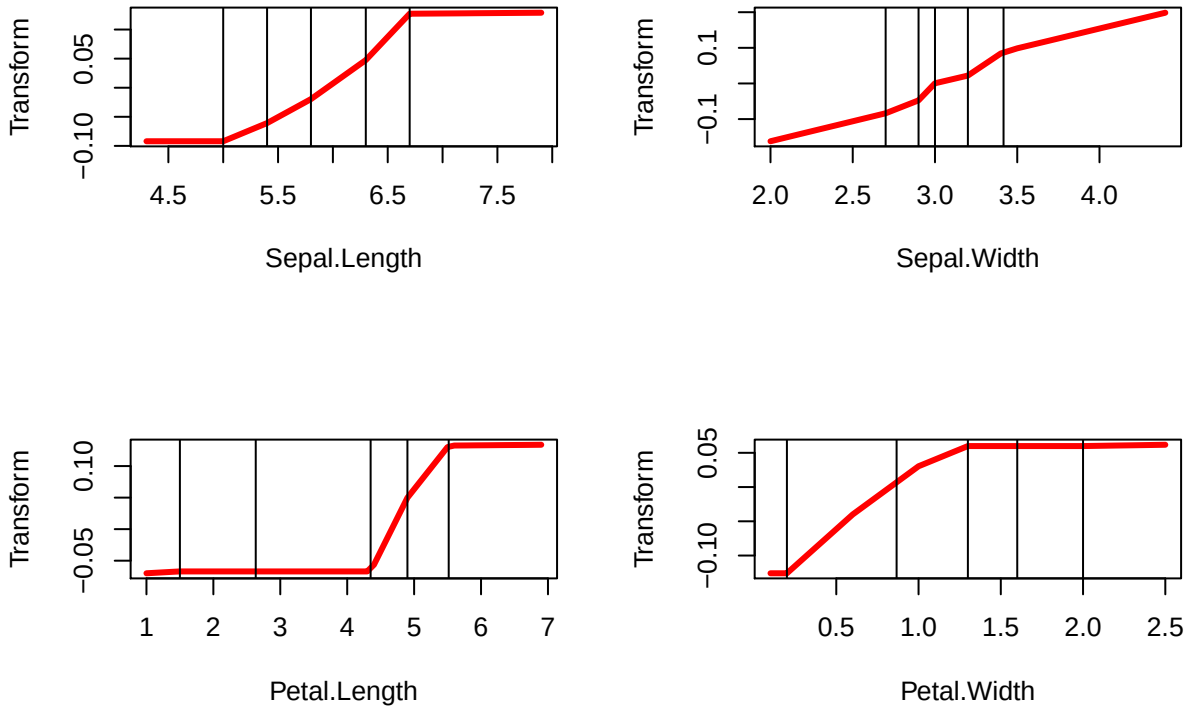


Figure 20: Iris Data, Transformations

Discriminant analysis decomposes the total dispersion matrix T into a sum of a between-groups dispersion B and a within-groups dispersion W, and then finds directions in the space spanned by the variables for which the between-variance is largest relative to the total variance. Homogeneity analysis optimizes the sum of the r largest eigenvalues of $T^{-1}B$. Before optimal transformation these eigenvalues for the iris data are 0.9698722, 0.2220266, after transformation they are 0.9789787, 0.7874823.

Multiset Canonical Correlation Example: Thirteen Personality Scales

This is the same example as before, but now we group the five scales from the Eysenck Personality Inventory and the five from the Big Five inventory into sets. The remaining three variables define three separate sets. No cpies are used, and we use monotone cubic splines with the interior knots at the quartiles.

```
epi_knots <- lapply (epi, function (x) fivenum (x)[2:4])
epi_degrees <- rep (3, 13)
epi_sets <- c(1,1,1,1,1,2,2,2,2,2,3,4,5)
```

```
h <- homals(epi, epi_knots, epi_degrees, epi_ordinal, epi_sets, epi_copies, verbose = FALSE)
```

In 196 iterations we find minimum loss 0.4724286. The object scores are in figure 21 and the transformation plots in figure 22.

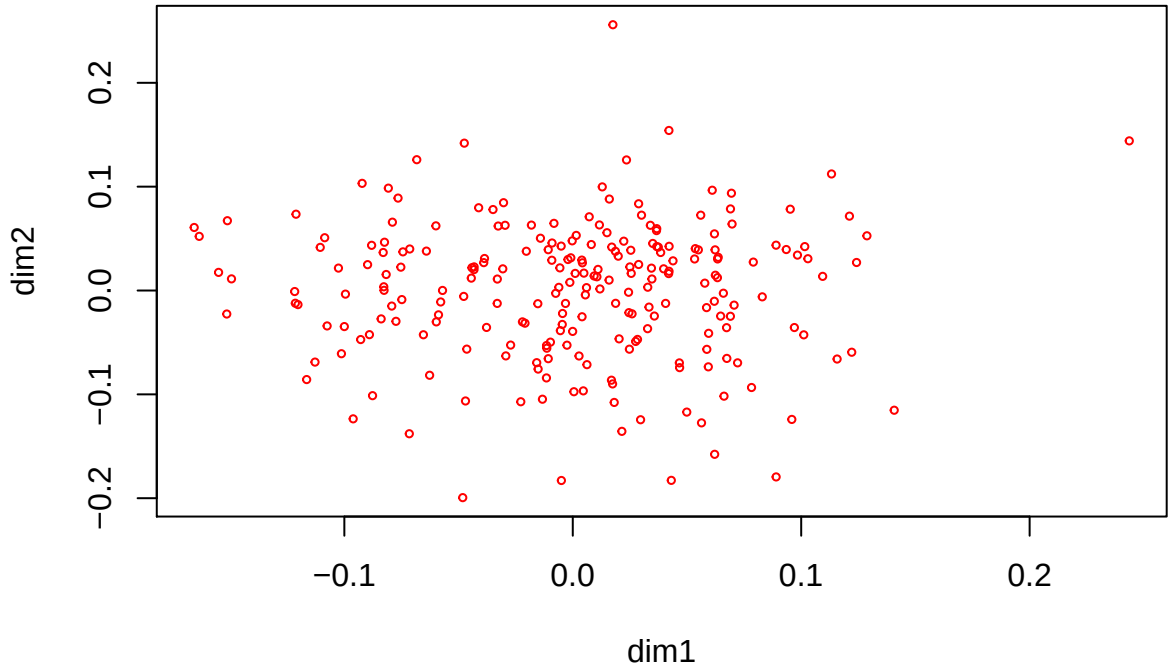
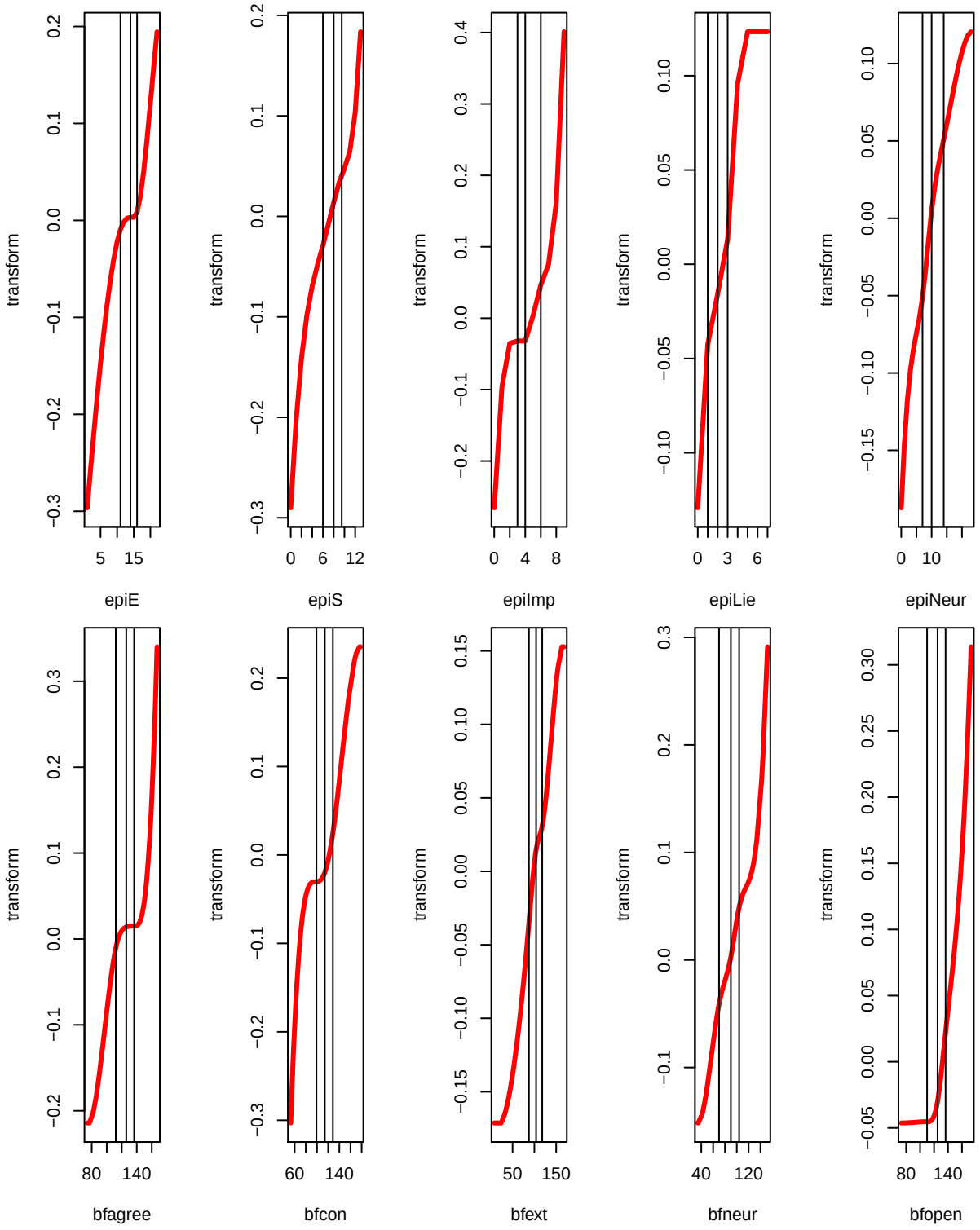


Figure 21: Personality Scales, Multi-Set, Objects Scores


transform
epiE
epiS
epiImp
epiLie
epiNeur
bfagree
bfcon
bfext
bfneur
bfopen

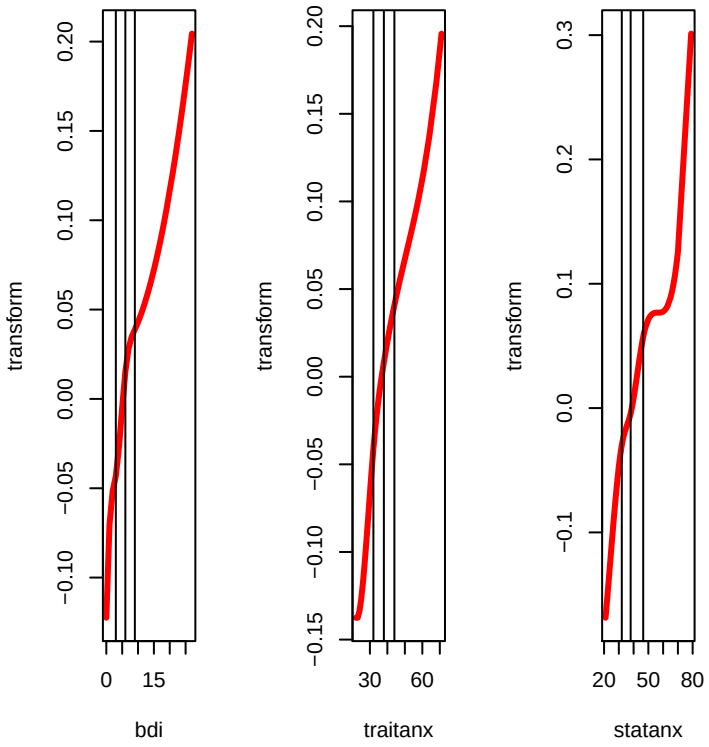


Figure 22: Personality Scales, Multi-Set, , Transformations

Appendix: Code

```
source("gs.R")
source("bsplineBasis.R")
library (nnls)

homals <-
  function (data,
            knots,
            degrees,
            ordinal,
            sets = 1:ncol(data),
            copies = rep (1, ncol(data)),
            ndim = 2,
            itmax = 1000,
            eps = 1e-6,
            seed = 123,
            verbose = TRUE) {
    nvars <- ncol (data)
    nobs <- nrow (data)
    nsets <- max (sets)
    ncops <- sum (copies)
    lmax <- ndim * nsets
    indices <- homalsIndices (nvars, nsets, ordinal, sets, copies)
    setup <-
      homalsSetup (data, knots, degrees, ordinal, sets , copies, indices$expand)
    initials <-
      homalsInitials (
```

```
      setup$q,
      setup$r,
      nobs,
      nsets,
      ncops,
      ndim,
      indices$expand,
      indices$nvarsets,
      indices$ordicops,
      seed
    )
  fold <- 0
  itel <- 1
  setcopies <- indices$setcopies
  h <- initials$h
  a <- initials$a
  x <- initials$x
  for (s in 1:nsets) {
    hh <- h[1:nobs, which (setcopies == s), drop = FALSE]
    aa <- a[which (setcopies == s), 1:ndim, drop = FALSE]
    fold <- fold + sum ((x - hh %*% aa) ^ 2)
  }
  fold <- fold / lmax
  repeat {
    xz <- matrix(0, nobs, ndim)
    fnew <- fmid <- 0
    for (s in 1:nsets) {
      id <- which (setcopies == s)
      hh <- h[1:nobs, id, drop = FALSE]
      lf <- lm.fit (hh, x)
      aa <- lf$coefficients
      rs <- lf$residuals
      kappa <- max (eigen (crossprod (aa))$values)
      fmid <- fmid + sum (rs ^ 2)
      target <- hh + tcrossprod (rs, aa) / kappa
      k <- 1
      for (l in id) {
        ql <- setup$q[[l]]
        vl <- setup$v[[l]]
        if (indices$ordicops[l]) {
          hz <- drop (crossprod (ql, target[, k]))
          ns <- nnls (vl, hz)
          rz <- coefficients (ns)
          hk <- drop (ql %*% (hz - drop (vl %*% rz)))
        }
        else {
          hk <- ql %*% crossprod (ql, target[, k])
        }
        hh[, k] <- hk / sqrt (sum (hk ^ 2))
        k <- k + 1
      }
      ha <- hh %*% aa
      xz <- xz + ha
```

```
        fnew <- fnew + sum ((x - ha) ^ 2)
        h[1:nobs, id] <- hh
        a[id, 1:ndim] <- aa
      }
      fmid <- fmid / lmax
      fnew <- fnew / lmax
      if (verbose)
        cat(
          "Iteration: ",
          formatC (itel, width = 3, format = "d"),
          "fold: ",
          formatC (
            fold,
            digits = 8,
            width = 12,
            format = "f"
          ),
          "fmid: ",
          formatC (
            fmid,
            digits = 8,
            width = 12,
            format = "f"
          ),
          "fnew: ",
          formatC (
            fnew,
            digits = 8,
            width = 12,
            format = "f"
          ),
          "\n"
        )
      if ((itel == itmax) || ((fold - fnew) < eps))
        break
      itel <- itel + 1
      fold <- fnew
      x <- gs (center (xz))$q
    }
    return (list (
      f = fnew,
      ntel = itel,
      x = x,
      a = a,
      h = h
    ))
  }

homalsIndices <- function (nvars, nsets, ordinal, sets, copies) {
  ordicops <- logical (0)
  expand <- numeric (0)
  for (j in 1:nvars) {
    ordicops <- c (ordicops, ordinal [j], rep (FALSE, copies[j] - 1))
```

```
    expand <- c (expand, rep (j, copies [j]))
  }
  setcopies <- sets[expand]
  nvarsets <- numeric(0)
  for (s in 1:nsets) {
    nvarsets <- c(nvarsets, sum (copies[which (sets == s)]))
  }
  return (list (
    ordicops = ordicops,
    setcopies = setcopies,
    expand = expand,
    nvarsets = nvarsets
  ))
}

homalsSetup <-
  function (data,
            knots,
            degrees,
            ordinal,
            sets,
            copies,
            expand) {
    nvars <- ncol (data)
    nobs <- nrow (data)
    ncops <- sum (copies)
    g <- b <- q <- r <- v <- list()
    for (l in 1:ncops) {
      j <- expand[l]
      mm <- is.na(data[, j])
      nm <- !mm
      dm <- data[nm, j]
      if (degrees[j] < 0) {
        gn <- ifelse (outer (dm, unique (dm), "=="), 1, 0)
      }
      else {
        gn <- bsplineBasis (dm, degrees[j], knots[[j]])
      }
      nr <- ncol (gn)
      ns <- sum (mm)
      gg <- matrix (0, nobs, nr + ns)
      gg[!mm, 1:nr] <- gn
      if (ns > 0) {
        gg[mm, nr + (1:ns)] <- diag (ns)
      }
      gg <- center (gg)[,-1, drop = FALSE]
      g <- c (g, list (gg))
      bb <- gs (g[[l]])
      q <- c(q, list (bb$q))
      r <- c (r, list (bb$r))
      vv <- makeV (data[[j]], q[[l]])
      v <- c (v, list (vv))
    }
```

```
    return (list (q = q, r = r, v = v))
  }

homalsInitials <-
  function (q,
            r,
            nobs,
            nsets,
            ncops,
            ndim,
            expand,
            nvarsets,
            ordicops,
            seed) {
    set.seed (seed)
    x <- matrix (rnorm (nobs * ndim), nobs, ndim)
    x <- gs (center (x))$q
    a <- matrix (rnorm (ncops * ndim), ncops, ndim)
    h <- matrix (0, nobs, ncops)
    for (l in 1:ncops) {
      ql <- q[[l]]
      rl <- r[[l]]
      if (ordicops[l])
        cf <- 1:ncol (ql)
      else
        cf <- rnorm (ncol (ql))
      h[, l] <- drop (ql %*% rl %*% cf)
    }
    h <- center (h)
    h <- apply (h, 2, function (z)
      z / sqrt (sum (z ^ 2)))
    return (list (x = x, a = a, h = h))
  }

makeV <- function (x, g) {
  r <- order (x)[1:length(which(!is.na(x)))]
  n <- length (r)
  m <- ncol (g)
  v <- numeric (0)
  k <- 0
  for (i in 1:(n - 1)) {
    ri <- r[i]
    rj <- r[i + 1]
    if (is.na(x[ri]) || is.na(x[rj]))
      next
    if (x[rj] > x[ri]) {
      v <- c(v, g[ri,] - g[rj,])
      k <- k + 1
    }
  }
  return (t(matrix (v, k, m, byrow = TRUE)))
}
```

```r
center <- function (x) {
  return (apply (x, 2, function (z)
    z - mean (z)))
}

normalize <- function (x) {
  return (x / sqrt (sum (x ^ 2)))
}
```

Appendix: NEWS

016: 06/28/15

- added figure captioning and numbering
- separated computation and plotting chunks
- hlines around figures
- rescaled small example
- added table of contents
- redid figures and computations in single chapter
- added NEWS section
- added link to pdf and Rmd versions

017: 06/29/15

- added nominal, ordinal, numerical
- added ALS algorithm for ordinal
- expanded NLPCA section

018: 06/30/15

- more ALS for ordinal
- worked on NLPCA example

019: 07/01/15

- minor editing changes throughout
- introduced QR decomposition of indicators earlier

020: 07/02/15

- transformation plots for epi example (still a bug in ordinal single)

021: 07/06/15

- squashed the bug in ordinal
- added TO DO appendix
- code now has inner iterations

022: 07/07/13

- added single numerical example
- some NLPCA theory added

023: 07/08/15

- center the basis matrices in the code

024: 07/08/15

- added star plots for MCA degree zero
- added graphics : yes and keep tex: yes to the YAML to please pandoc
- material on sets of variables

025: 07/10/15

- material on copies

027: 07/11/15

- some editing
- copies algorithm

028: 07/12/15

- majorization part of copies algorithm

029: 07/13/15

- majorization part of copies algorithm

031: 07/21/15

- new copies-based code wrapped up and inserted
- redid all examples and plots with the new homals program
- made ftp directory

032: 07/21/15

- added neumann example

033: 07/21/15

- minor edits
- normalized loss

034: 07/22/15

- extensive editing
- itermax default to 1000
- got rid of details of "old" algorithm
- introduce copies much earlier

035: 07/24/15

- (X,H,A) loss replacing (X,Y) loss earlier
- transformation plots become line plots
- added stepPlotter function for discrete case
- redid discrete plots with stepPlotter

036: 07/27/15

- described copies algorithm earlier and in more detail
- added caches for homals solutions

037: 08/02/15

- added seed to homals parameter list for homalsInitials
- fixed homalsSetup bug for dim(g[[j]])[2]=2 (added DROP=FALSE)

038: 08/03/15

- various edits
- move copies loss even more to the beginning
- number algorithms

039: 08/04/15

- added iris example

040: 08/05/15

- added multiset CCA example

041: 08/19/15

- complete rewrite, remove old notation, Gifi approach

042: 08/20/15

- still working on the copies rewrite

043: 09/27/15

- reorganizing and rewriting

044: 10/01/15

- reorganizing and rewriting

045: 10/15/15

- reorganizing and rewriting

046: 10/20/15

- reorganizing and rewriting

047: 10/21/15

- implementation details added

048: 10/23/15

- cut out some unnecessary stuff

049: 10/25/15

- special cases expanded
- loss geometry eliminated for now

050: 11/01/15

- needs more testing, but added primary/secondary ties to ordinal
- may not work yet, but provisions for missing data

051: 11/02/15

- missing data implemented, needs more testing

052: 11/05/15

- some edits
- degree = -1 now means categorical variable, makes indicator without calling B-spline code
- primary/secondary ties for each variable separately

053: 11/07/15

- eliminate secondary approach to ties, which does not make much sense

054: 11/08/15

- some code changes to deal with categorical variables
- reformat code

100: 11/10/15

- some final edits

101: 11/10/15

- added some before and after statistics

102: 11/12/15

- lots of small edits

References

Carroll, J.D. 1968. "A Generalization of Canonical Correlation Analysis to Three or More Sets of Variables." In *Proceedings of the 76th Annual Convention of the American Psychological Association*, 227–28. Washington, D.C.: American Psychological Association.

Cordier, B. 1965. "L'Analyse Factorielle des Correspondances." Thèse de Troisieme Cycle, Université de Rennes; Faculté des Sciences.

De Leeuw, J. 2015. "Exceedingly Simple B-Spline Code." http://rpubs.com/deleeuw/79161.

De Leeuw, J. 1968. "Canonical Discriminant Analysis of Relational Data." Research Note 007-68. Department of Data Theory FSW/RUL. http://deleeuwpdx.net/janspubs/1968/reports/deleeuw_R_68e.pdf.

———. 1974. *Canonical Analysis of Categorical Data*. Leiden, The Netherlands: Psychological Institute, Leiden University. http://deleeuwpdx.net/janspubs/1974/books/deleeuw_B_74.pdf.

———. 1975. "A Normalized Cone Regression Approach to Alternating Least Squares Algorithms." Department of Data Theory FSW/RUL. http://deleeuwpdx.net/janspubs/1975/notes/deleeuw_U_75a.pdf.

———. 1994. "Block Relaxation Algorithms in Statistics." In *Information Systems and Data Analysis*, edited by H.H. Bock, W. Lenski, and M.M. Richter, 308–24. Berlin: Springer Verlag. http://deleeuwpdx.net/janspubs/1994/chapters/deleeuw_C_94c.pdf.

———. 2014. "History of Nonlinear Principal Component Analysis." In *The Visualization and Verbalization of Data*, edited by J. Blasius and M. Greenacre. Chapman; Hall. http://deleeuwpdx.net/janspubs/2014/chapters/deleeuw_C_14.pdf.

———. 2015a. "Aspects of Correlation Matrices." http://deleeuwpdx.net/pubfolders/aspect/aspect.pdf.

———. 2015b. "Regression with Linear Inequality Restrictions on Predicted Values." http://rpubs.com/deleeuw/78897.

De Leeuw, J., and P. Mair. 2009. "Homogeneity Analysis in R: The Package Homals." *Journal of Statistical Software* 31 (4): 1–21. http://deleeuwpdx.net/janspubs/2009/articles/deleeuw_mair_A_09a.pdf.

De Leeuw, J., and J.L.A. Van Rijckevorsel. 1980. "HOMALS and Princals: Some Generalizations of Principal Components Analysis." In *Data Analysis and Informatics*. Amsterdam: North Holland Publishing Company. http://deleeuwpdx.net/janspubs/1980/chapters/deleeuw_vanrijckevorsel_C_80.pdf.

De Leeuw, J., J.L.A. Van Rijckevorsel, and H. Van der Wouden. 1981. "Nonlinear Principal Component Analysis Using B-Splines." *Methods of Operations Research* 43: 379–94. http://deleeuwpdx.net/janspubs/1981/articles/deleeuw_vanrijckevorsel_vanderwouden_A_81.pdf.

Gibson, W.A. 1962. "On the Least Squares Orthogonalization of an Oblique Transformation." *Psychometrika* 27: 193–95.

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

Gower, J.C., and D.J. Hand. 1996. *Biplots*. Monographs on Statistics and Applied Probability 54. Chapman; Hall.

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.

Heiser, W.J. 1995. "Convergent Computing by Iterative Majorization: Theory and Applications in Multidimensional Data Analysis." In *Recent Advantages in Descriptive Multivariate Analysis*, edited by W.J.

Krzanowski, 157–89. Oxford: Clarendon Press.

Lange, K., D.R. Hunter, and I. Yang. 2000. "Optimization Transfer Using Surrogate Objective Functions." *Journal of Computational and Graphical Statistics* 9: 1–20.

Meulman, J.J., and W.J. Heiser. 2011. *IBM SPSS Categories 20.* IBM Corporation.

Michailidis, G., and J. De Leeuw. 1998. "The Gifi System for Descriptive Multivariate Analysis." *Statistical Science* 13: 307–36. http://deleeuwpdx.net/janspubs/1998/articles/michailidis_deleeuw_A_98.pdf.

Revelle, William. 2015. *Psych: Procedures for Psychological, Psychometric, and Personality Research.* Evanston, Illinois: Northwestern University. http://CRAN.R-project.org/package=psych.

Tenenhaus, A., and M. Tenenhaus. 2011. "Regularized Generalized Canonical Correlation Analysis." *Psychometrika* 76: 257–84.

Van der Burg, E., J. De Leeuw, and R. Verdegaal. 1988. "Homogeneity Analysis with K Sets of Variables: An Alternating Least Squares Approach with Optimal Scaling Features." *Psychometrika* 53: 177–97. http://deleeuwpdx.net/janspubs/1988/articles/vanderburg_deleeuw_verdegaal_A_88.pdf.

Van der Velden, M. 2012. "On Generalized Canonical Correlation Analysis." In *Proceedings 58th World Statistical Congress, 2011, Dublin*, 758–65. The Hague: International Statiatical Instutute.

Van der Velden, M., and Y. Takane. 2012. "Generalized Canonical Correlation Analysis with Missing Values." *Computational Statistics* 27: 551–71.

Van Rijckevorsel, J.L.A., and J. De Leeuw, eds. 1988. *Component and Correspondence Analysis.* Wiley. http://deleeuwpdx.net/janspubs/1988/books/vanrijckevorsel_deleeuw_B_88.pdf.

Wilson, E.B. 1926. "Empiricism and Rationalism." *Science* 64: 47–57.