Metric Multidimensional Unfolding

Jan de Leeuw

March 4, 2026

We review Schönemann's technique to reconstruct a configuration of $n + m$ points from an $n \times m$ off-diagonal submatrix of the distance matrix. The technique is meant for error-free distances, but we pat spacial attention to what happens if it is applied to dissimilarities which may not have a perfect representation in Euclidean space. Examples are analyzed and R programs are provided.

Table of contents

Note: This is a working manuscript which can be expanded/updated multiple times. All suggestions for improvement are welcome. All Rmd, tex, html, pdf, R, and C files are in the public domain. Attribution will be appreciated, but is not required. The files can be found at https://github.com/deleeuw/schoenemann

1 Introduction

At the core of classical metric Multidimensional Scaling (MDS) is the Schoenberg-Young-Householder-Torgerson-Gower (SYHTG) Theorem. It was first proved in a distance geometry context by Schoenberg (1935) and then (independently, it seems) by Young and Householder (1938) for application in the very first versions of MDS in psychometrics. It was given its current more computational form by Torgerson (1958) and Gower (1966).

Suppose D is a square, symmetric, and hollow non-negative matrix with dissimilarities between n objects. Define

$$C := -\frac{1}{2}J_n(D \times D)J_n \tag{1}$$

where $J_n = I - n^{-1}e_n e_n'$ is the centering matrix, e_n has all its elements equal to one, and $D \times D$ is the elementwise square of D. The SYHTG theorem says that there is an $n \times p$ matrix X such that

$$d_{ij} = \sqrt{\sum_{s=1}^{p}(x_{is} - x_{js})^2} \tag{2}$$

if and only if C is positive semidefinite with rank $r \leq p$. Moreover in that case $n \times r$ solutions X can be found from any full rank decomposition $C = XX'$.

The SYHTG result has an existence part, which says that a Euclidean representation exists if and only if the matrix C is positive semi-definite. This is an algebraic result, which is of limited direct relevance for data analysis. In practical situations data will be fallible, and we merely expect that dissimilarities can be approximated by Euclidean distances. The second part of the SYHTG result is an approximation result, which says that a p-dimensional Euclidean configuration of points with distances that approximate the given dissimilarities can be found by using the p largest eigenvalues of C and the corresponding eigenvectors.

Multidimensional Unfolding (MDU) is a special case of MDS in which there are two disjoint sets of objects. The first set has n elements, the second set has m elements. The non-negative matrix D is $n \times m$ and contains the dissimilarities between the elements of the first and second set. The within-set dissimilarities are missing. There were some tentative attempts to generalize the SYHTG theorem to off-diagonal data of this form, but not much was achieved until the work of Schönemann (1970), which also has a review of the accomplishments until 1970.

Schönemann's technique is developed for perfect data, i.e. for dissimilarities that can be faithfully represented as distances in Euclidean space. It is an exact algebraic method that does not involve any approximation, similar to the embedding part of the SYHTG theorem. In the second part of his paper Schönemann applies the embedding calculations to "fallible" data, using some small artificial examples. Unlike the SYHTG approximation part, which has a

clear least squares interpretation, these calculations proceed by what Joe Kruskal used to call "ignoring errors".

Heiser and De Leeuw (1979) present various variations of the Schönemann technique, and review the literature until 1979. They also compare the results on various examples with metric MDS results obtained from smacof, the standard iterative least squares MDS technique. And they provide an example of applying their Schönemann variations on a real data analysis example. Although the standard MDS textbook of Borg and Groenen (2005) pays a lot of attention to unfolding, the work of Schönemann is only mentioned in a single footnote.

2 Perfect Data

Suppose D is an $n \times m$ matrix with non-negative numbers. The problem we study in this paper is solving the system of equations

$$d_{ij} = \sum_{s=1}^{p} (x_{is} - y_{js})^2. \tag{3}$$

for an $n \times p$ matrix X and an $m \times p$ matrix Y. In matrix notation (3) becomes

$$D = \alpha e_m' + e_n \beta' - 2XY', \tag{4}$$

with $\alpha := \text{diag}(XX')$ and $\beta := \text{diag}(YY')$ and with e_n and e_m vectors of appropriate length with all elements equal to one. In this section we study the case in which we have "perfect data", i.e. in which (4) has a solution (X, Y) for some p. Our problem is to recover X and Y from D.

Suppose $J_n = I - n^{-1} e_n e_n'$ and $J_m = I - m^{-1} e_m e_m'$ are centering matrices. Then

$$C := -\frac{1}{2} J_n D J_m = J_n X Y' J_m. \tag{5}$$

Suppose C has rank p and $C = GH'$ is any full-rank decomposition of C. Being able to construct X and Y from C assumes that $J_n X$ and $J_m Y$ have the same rank as X and Y, which means that we assume that e_n is not in the column-space of X and e_m is not in the column space of Y. If that assumption is violated the full-rank decomposition has rank $q < p$. In an extreme case, suppose $Y = e_m y'$ for some p-vector y. Then all rows of D are the same and $C = 0$.

From (5) it follows that there exist $p \times p$ nonsingular matrices T and S and vectors μ and ν of length p such that

$$X = GT + e_n\mu', \tag{6a}$$
$$Y = HS + e_m\nu', \tag{6b}$$
$$TS' = I. \tag{6c}$$

Consequently

$$XX' = GMG' + GT\mu e_n' + e_n\mu'T'G' + \mu'\mu\, e_n e_n', \tag{7a}$$
$$YY' = HNH' + HS\nu e_m' + e_m\nu'S'H' + \nu'\nu\, e_m e_m', \tag{7b}$$
$$XY' = GH' + GT\nu e_m' + e_n\mu'S'H' + \mu'\nu\, e_n e_m', \tag{7c}$$

where $M := TT'$ and $N := SS'$. Note that $N = M^{-1}$.

Define an $n \times \frac{1}{2}p(p+1)$ matrix $G \star G$ which has a column for each pair of indices $1 \le s \le t \le p$. Columns of $G \star G$ are elementwise products of columns of G according to

$$(G \star G)_{(s,t)} := \begin{cases} 2g_s g_t & \text{if } s \ne t, \\ g_s^2 & \text{if } s = s. \end{cases} \tag{8}$$

Matrix $H \star H$ is defined in the same way, using the colums of H. Next define the $\frac{1}{2}p(p+1)$-element vectors $\overline{m}$ and $\vec{n}$, which have the diagonal and subdiagonal elements of M and N, in the same order as the columns of $G \star G$ and $H \star H$. Then $\mathrm{diag}(GMG') = (G \star G)\overline{m}$ and $\mathrm{diag}(HNH') = (H \star H)\vec{n}$.

In addition to (7a)-(7c) we have

$$\alpha e_m' = (G \star G)\overline{m}e_m' + 2GT\mu e_m' + \mu'\mu e_n e_m', \tag{9a}$$
$$e_n\beta' = e_n\vec{n}'(H \star H)' + 2e_n\nu'S'H' + \nu'\nu e_n e_m'. \tag{9b}$$

Substituting (7a)-(7c) and (9a)-(9b) in equation (4) gives

$$D = -2GH' + 2GT\eta e_m' - 2e_n\eta'S'H' + (G \star G)\overline{m}e_m' + e_n\vec{n}'(H \star H)' + \eta'\eta e_n e_m', \tag{10}$$

with $\eta := \mu - \nu$.

We now take what could be described as a "row-oriented approach", by eliminating all references to H and S from equation (10). Define $F := D + 2GH'$ and $\tilde{F} := J_n F$. Then, from (10),

$$\tilde{F} = 2J_n GT\eta e_m' + J_n(G \star G)\overline{m}e_m' \tag{11}$$

Since $C = GH'$ it follows that $D + 2GH' = D - J_n D J_m$ and all m columns of $\tilde{F}$ are identical to the centered row-averages of D.

Assume $\tilde{F}$ is non-zero, i.e. that not all row-sums of D are the same. Take any of its columns, name it f. Define a matrix

$$K := \begin{bmatrix} 2J_n G & J_n(G \star G) \end{bmatrix}. \tag{12}$$

We must have

$$f = \begin{bmatrix} 2J_n G & J_n(G \star G) \end{bmatrix} \begin{bmatrix} T\eta \\ \overrightarrow{m} \end{bmatrix}. \tag{13}$$

This is a system of n linear equations in $\frac{1}{2}p(p+3)$ unknowns. Suppose it has a unique solution for $T\eta$ and $\overrightarrow{m}$. If $n \geq \frac{1}{2}p(p+3)$ then the solution can be computed by inverting any non-singular submatrix of order $\frac{1}{2}p(p+3)$. No matter how we choose the submatrix, the solution will be the same. From the solution of (13) use the elements of $\overrightarrow{m}$ to reconstruct M. Use any decomposition method to find T from $M = TT'$ and use T to find η and $S = (T')^{-1}$. Assume without loss of generality that $\nu = 0$. Then $Y = HS$ and $X = GT + e_n \mu'$ gives a solution, with Y column-centered.

If the system of equations (4) has a solution, then the linear equations (13) are a consequence of (4), and thus have at least one solution with a positive definite matrix M. If (13) has more than one solution, because the matrix K has rank less than $\frac{1}{2}p(p+3)$, then we must find a solution for which M is positive definite. Conversely, if (13) has a solution with M positive definite then system (4) has a solution with dimension $p = \text{rank}(C)$.

Note that it is also possible to go the "column-oriented" way, by defining $\tilde{F}$ as FJ_m, which now has identical rows depending on N and S. This also leads to a solution for X and Y, which in the perfect case should be identical to the "row-oriented" one. We went the "row-oriented" way because we have the practical situations in mind in which n is larger, often much larger, than m. And we can always find the "column-oriented" solution by applying the "row-oriented" calculations to the transpose of D. For example, if all row-sums of D are the same then $\tilde{F} = 0$ and our "row-oriented" technique fails. In this case we can analyze the transpose of D instead, which may have non-constant row-sums.

3 Artificial Example

This section has in-line R code and R output for a 9×3 artificial example. In the code we use the same names (in lower case) for the key matrices as in our formulas (in upper case). The full-rank decomposition of C is computed from its singular value decomposition. Note that the solutions, which are $\hat{X}$ and $\hat{Y}$ in the output, are not the same as X and Y used to generate the data because of the rotational and translational indeterminacy. But the distances $\hat{D}$ between $\hat{X}$ and $\hat{Y}$ are the same as the distances D between X and Y, which is what matters.

```
x <- matrix(c(1:9, 1, 2, 3, 1, 2, 3, 1, 2, 3), 9, 2) / 10
y <- matrix(c(1, 1, -2, 1, 0, -1), 3, 2)
matrixPrint(x, digits = 2, width = 4)
```

```
     [,1]  [,2]
 [1,] +0.10 +0.10
 [2,] +0.20 +0.20
 [3,] +0.30 +0.30
 [4,] +0.40 +0.10
 [5,] +0.50 +0.20
 [6,] +0.60 +0.30
 [7,] +0.70 +0.10
 [8,] +0.80 +0.20
 [9,] +0.90 +0.30
```

```
matrixPrint(x, digits = 2, width = 4)
```

```
     [,1]  [,2]
 [1,] +0.10 +0.10
 [2,] +0.20 +0.20
 [3,] +0.30 +0.30
 [4,] +0.40 +0.10
 [5,] +0.50 +0.20
 [6,] +0.60 +0.30
 [7,] +0.70 +0.10
 [8,] +0.80 +0.20
 [9,] +0.90 +0.30
```

```
a <- diag(tcrossprod(x))
b <- diag(tcrossprod(y))
d <- outer(a, b, "+") - 2 * tcrossprod(x, y)
matrixPrint(d, digits = 4, width = 6)
```

```
      [,1]    [,2]    [,3]
 [1,] +1.6200 +0.8200 +5.6200
 [2,] +1.2800 +0.6800 +6.2800
 [3,] +0.9800 +0.5800 +6.9800
 [4,] +1.1700 +0.3700 +6.9700
 [5,] +0.8900 +0.2900 +7.6900
 [6,] +0.6500 +0.2500 +8.4500
 [7,] +0.9000 +0.1000 +8.5000
 [8,] +0.6800 +0.0800 +9.2800
 [9,] +0.5000 +0.1000 +10.1000
```

```
c <- -(jmat(9) %*% d %*% jmat(3)) / 2
matrixPrint(c, digits = 4, width = 6)
```

```
      [,1]    [,2]    [,3]
 [1,] -0.5000 -0.4000 +0.9000
 [2,] -0.3000 -0.3000 +0.6000
 [3,] -0.1000 -0.2000 +0.3000
 [4,] -0.2000 -0.1000 +0.3000
 [5,] +0.0000 -0.0000 +0.0000
 [6,] +0.2000 +0.1000 -0.3000
 [7,] +0.1000 +0.2000 -0.3000
 [8,] +0.3000 +0.3000 -0.6000
 [9,] +0.5000 +0.4000 -0.9000
```

```
sv <- svd(c)
e <- diag(sqrt(sv$d[1:2]))
g <- sv$u[, 1:2] %*% e
matrixPrint(g, digits = 4, width = 6)
```

```
      [,1]    [,2]
 [1,] -0.7781 -0.0712
 [2,] -0.5174 +0.0723
```

```
 [3,] -0.2568 +0.2157
 [4,] -0.2606 -0.1435
 [5,] -0.0000 +0.0000
 [6,] +0.2606 +0.1435
 [7,] +0.2568 -0.2157
 [8,] +0.5174 -0.0723
 [9,] +0.7781 +0.0712
```

```
h <- sv$v[, 1:2] %*% e
matrixPrint(h, digits = 4, width = 6)
```

```
     [,1]    [,2]
[1,] +0.6178 +0.2718
[2,] +0.5401 -0.2842
[3,] -1.1579 +0.0124
```

```
gg <- cbind(g[, 1]^2, 2 * g[, 1] * g[, 2], g[, 2]^2)
hh <- cbind(h[, 1]^2, 2 * h[, 1] * h[, 2], h[, 2]^2)
f <- jmat(9) %*% (d + 2 * tcrossprod(g, h))
matrixPrint(f, digits = 4, width = 6)
```

```
      [,1]    [,2]    [,3]
 [1,] -0.3433 -0.3433 -0.3433
 [2,] -0.2833 -0.2833 -0.2833
 [3,] -0.1833 -0.1833 -0.1833
 [4,] -0.1933 -0.1933 -0.1933
 [5,] -0.0733 -0.0733 -0.0733
 [6,] +0.0867 +0.0867 +0.0867
 [7,] +0.1367 +0.1367 +0.1367
 [8,] +0.3167 +0.3167 +0.3167
 [9,] +0.5367 +0.5367 +0.5367
```

```
jg <- jmat(9) %*% g
jgg <- jmat(9) %*% gg
k <- cbind(2 * jg, jgg)
fm <- apply(f, 1, mean)
cof <- qr.solve(k, fm)
m <- matrix(c(cof[3], cof[4], cof[4], cof[5]), 2, 2)
```

```
t <- t(chol(m))
mn <- solve(t, cof[1:2])
xhat <- g %*% t + matrix(mn , 9, 2, byrow = TRUE)
matrixPrint(xhat, digits = 4, width = 6)
```

```
      [,1]    [,2]
 [1,] +0.1132 +0.0848
 [2,] +0.2264 +0.1695
 [3,] +0.3396 +0.2543
 [4,] +0.4102 +0.0421
 [5,] +0.5234 +0.1268
 [6,] +0.6366 +0.2116
 [7,] +0.7071 -0.0006
 [8,] +0.8203 +0.0841
 [9,] +0.9335 +0.1689
```

```
s <- t(solve(t))
yhat <- h %*% s
matrixPrint(yhat, digits = 4, width = 6)
```

```
     [,1]    [,2]
[1,] +1.1321 +0.8475
[2,] +0.9898 -0.1423
[3,] -2.1220 -0.7052
```

```
ahat <- diag(tcrossprod(xhat))
bhat <- diag(tcrossprod(yhat))
dhat <- outer(ahat, bhat, "+") - 2 * tcrossprod(xhat, yhat)
matrixPrint(dhat, digits = 4, width = 6)
```

```
      [,1]    [,2]    [,3]
 [1,] +1.6200 +0.8200 +5.6200
 [2,] +1.2800 +0.6800 +6.2800
 [3,] +0.9800 +0.5800 +6.9800
 [4,] +1.1700 +0.3700 +6.9700
 [5,] +0.8900 +0.2900 +7.6900
 [6,] +0.6500 +0.2500 +8.4500
 [7,] +0.9000 +0.1000 +8.5000
 [8,] +0.6800 +0.0800 +9.2800
 [9,] +0.5000 +0.1000 +10.1000
```

4 Imperfect Data

For real data the system (4) will generally not have a perfect solution for any p (i.e. there is no Euclidean configuration for which the D are squared distances). At various steps our technique necessarily differs from the technique in the perfect case.

First we have to pick a p. This can be done a priori, for instance by using the MDS default $p = 2$. Or we can pick p by eye-balling the singular values of C and try to find an "elbow" familiar from factor analysis. Instead of a full rank decomposition of C we then use a rank p approximation GH', for example by using the p dominant singular values and their singular vectors.

Now proceed with the calculations as in the perfect case. For imperfect data with $GH' \neq C$ the columns of F from (4) are not identical any more, and not equal to the centered row-means. So picking different columns will give different results. In some cases it may make sense to choose f equal to the row-averages of F. But note that, no matter how good or bad the fit of GH' to C is, the row-averages of F are always equal to the row-averages of D. There are other options, as long as they give the correct result in the perfect case. Thus picking the row-wise median or minimum or maximum is also a possibility. Remember that a rule that makes f small (in norm) will make M, and consequently T and X, small and will make S and Y large.

In some cases we can actually use the fact that the columns of F are different to perform m analyses, using each column of F in turn. The m different results then give some information about the fit and the stability of the solution.

In the next step of the technique we may find that system (13) is overdetermined and not solvable, or singular with multiple solutions. The usual case will be that no exact solution exists. As a compromise in that case we can use the Moore-Penrose inverse to compute the minimum norm least squares solution.

And finally, and this is the most troublesome problem, we may find that M constructed from the least squares solution is not positive definite. In that case case we cannot find a T such that $M = TT'$ and we are stuck. As this happens, for now, we give up.

Of course these problems will be less serious if the fit is good, i.e. if the elements of D are approximately squared Euclidean distances.

5 Real Examples

5.1 Airline Distances

Our first real example is a matrix of airline distances between 18 cities. The cities are

```
      [,1]
 [1,] Beijing
 [2,] Cape Town
 [3,] Hong Kong
 [4,] Honolulu
 [5,] London
 [6,] Melbourne
 [7,] Mexico
 [8,] Montreal
 [9,] Moscow
[10,] New Delhi
[11,] New York
[12,] Paris
[13,] Rio de Janeiro
[14,] Rome
[15,] San Francisco
[16,] Singapore
[17,] Stockholm
[18,] Tokyo
```

We start by computing the metric MDS solution on the full matrix.

The following two plots give the configuration and the D-Dhat plot for the airline data. The East-West and North-South dimensions show up clearly in the plot.

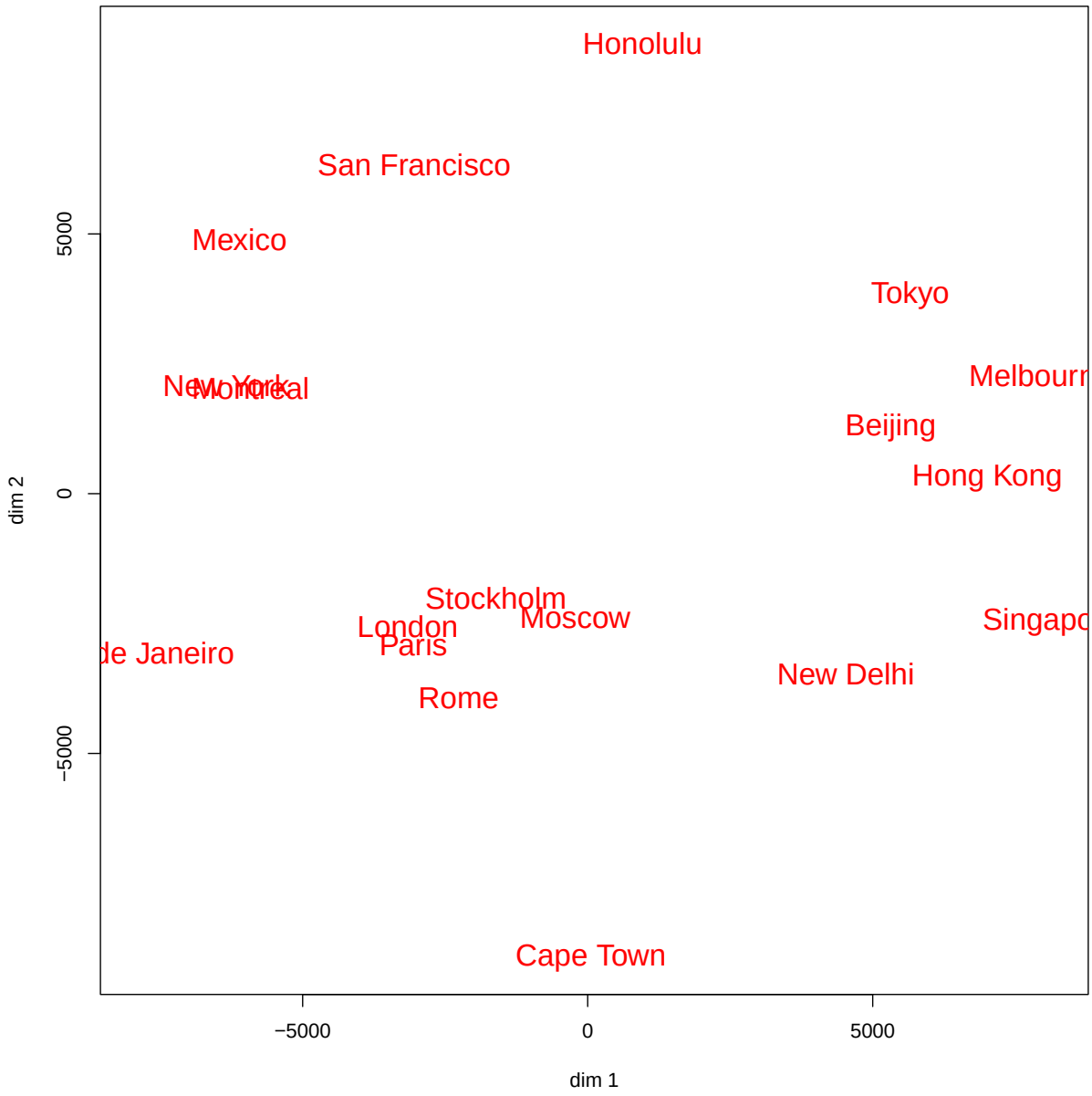

Figure 1: Airline Distances, Configuration Plot, Complete Data

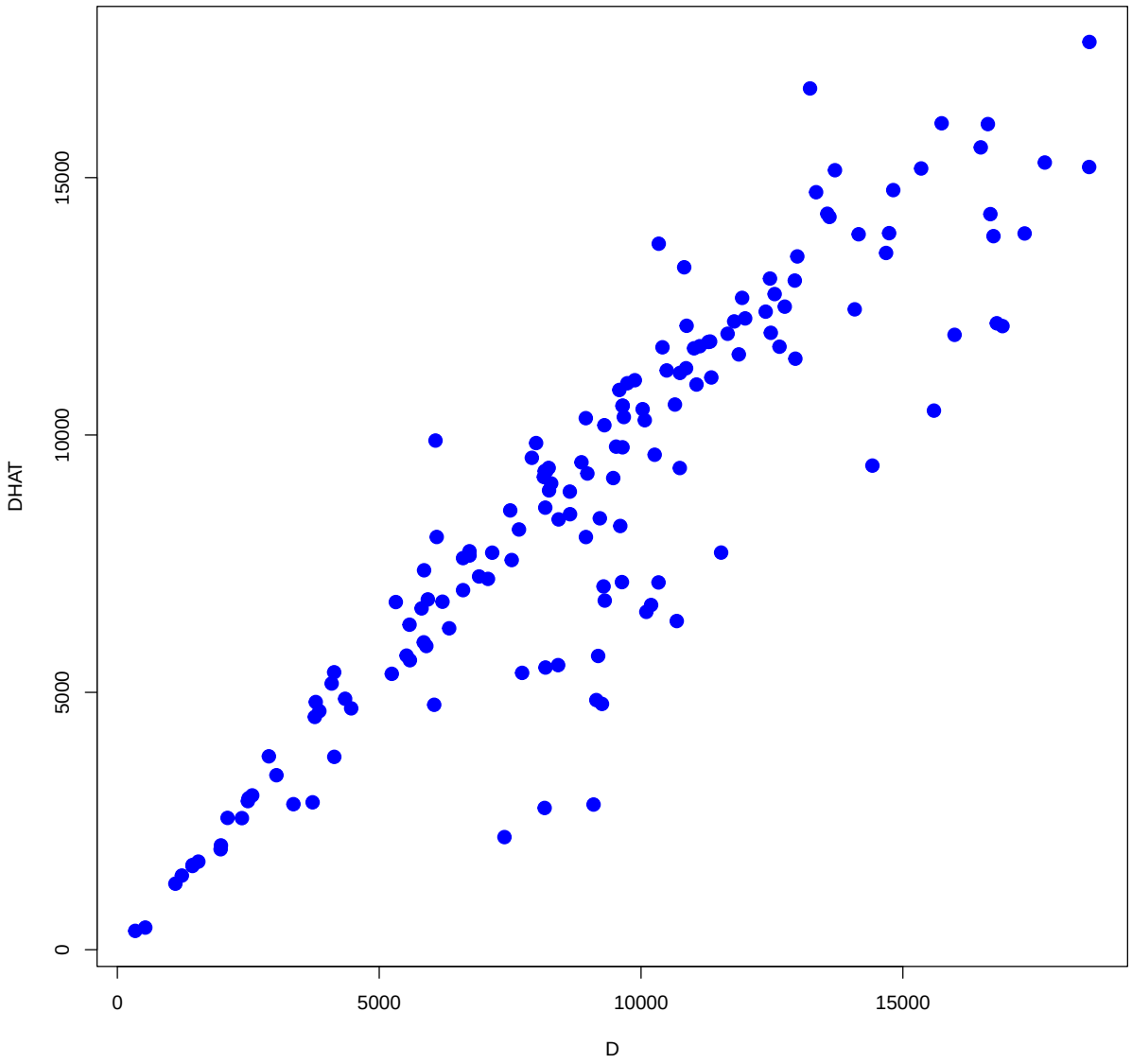

Figure 2: Airline Distances, Dist-Dhat Plot, Complete Data

We now create a 9×9 off-diagonal submatrix by using the even-numbered cities as rows and the rest as columns. The transpose of the submatrix is also analyzed. Although the two dimensions from the complete solution are still there the plots have some distortions

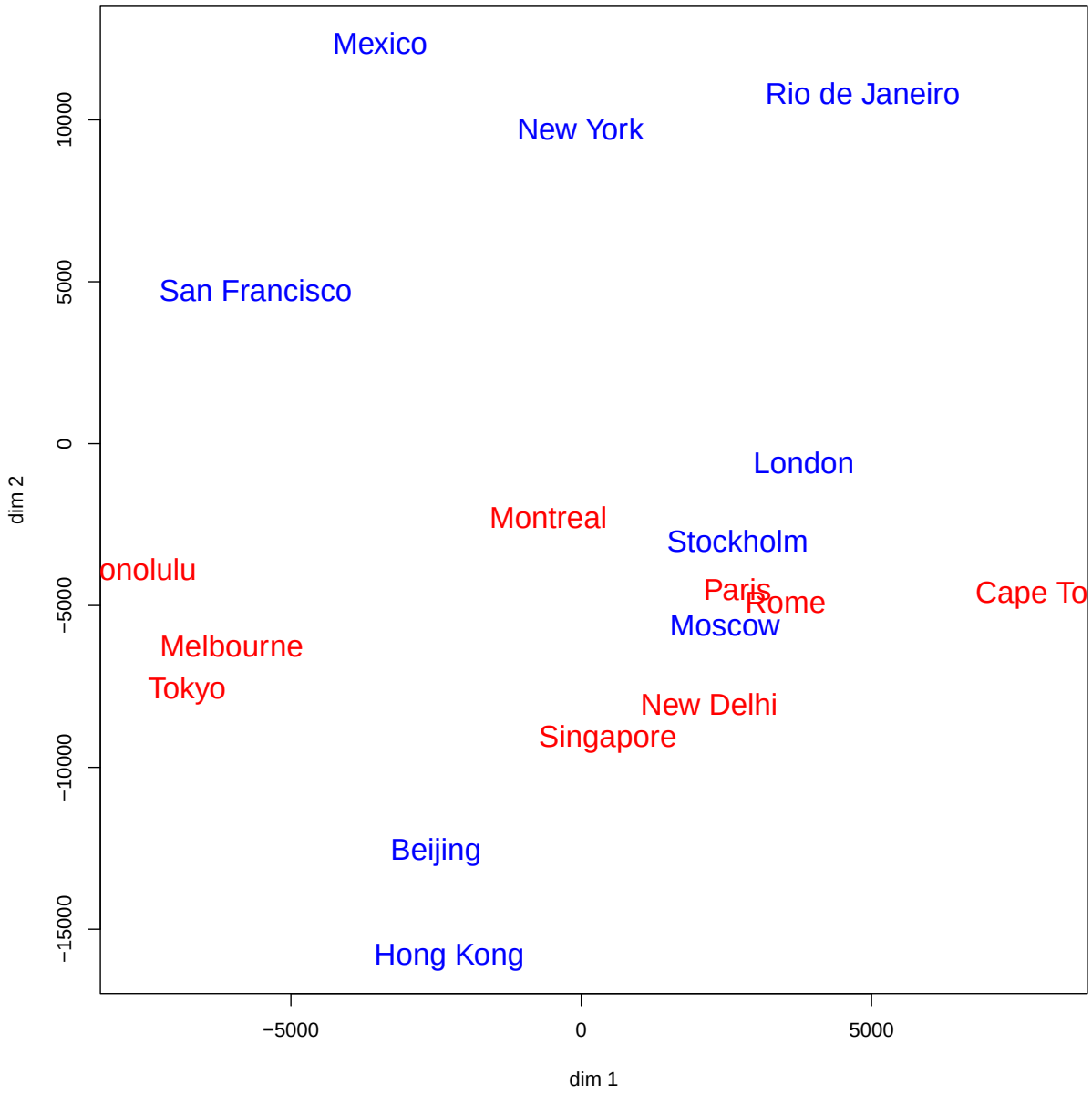

Figure 3: Airline Distances, Configuration Plot, Off-diagonal Submatrix

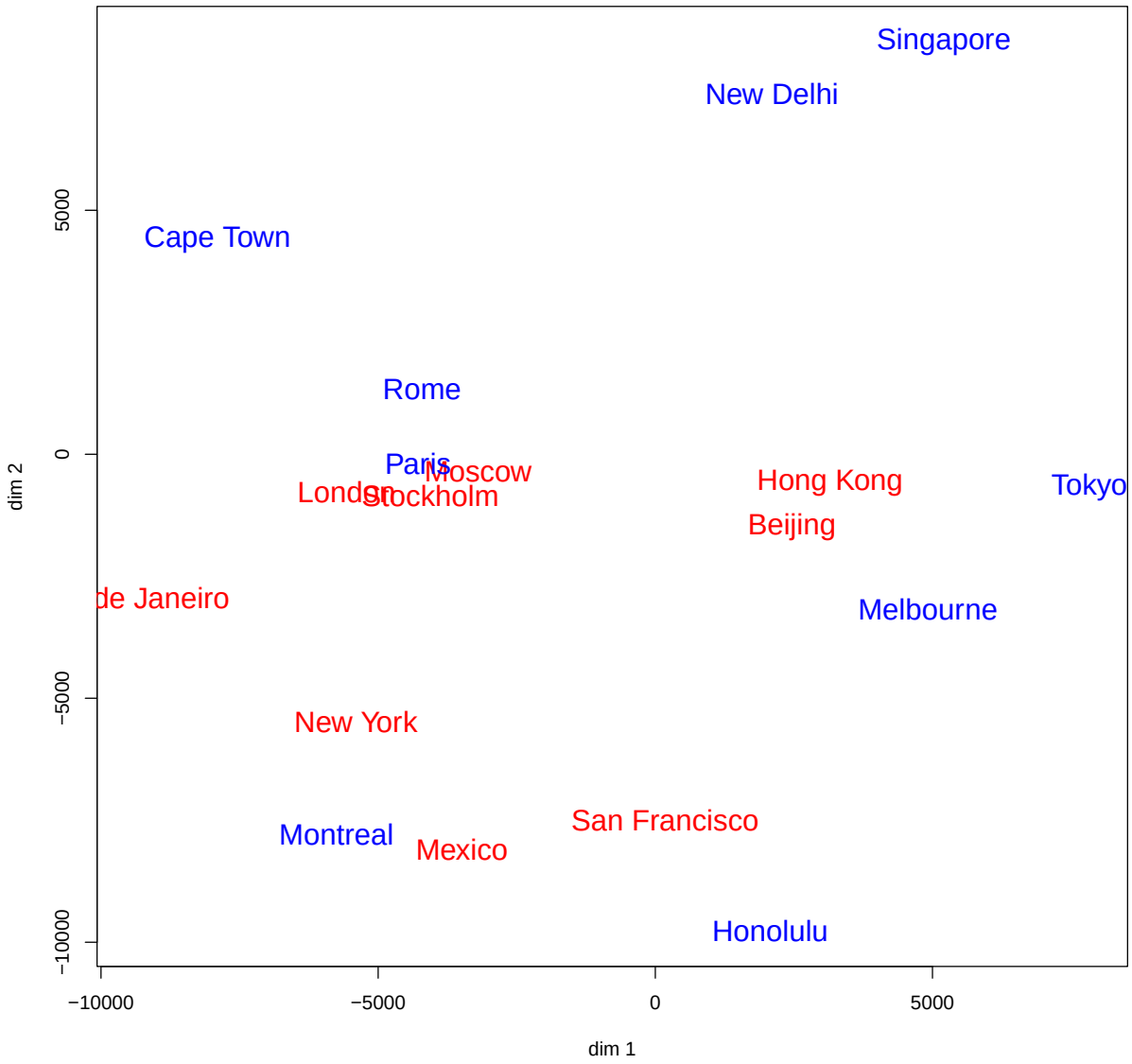

Figure 4: Airline Distances, Configuration Plot, Off-diagonal Submatrix, Transposed

5.2 Gold

Our next example is the same as the one in Heiser and De Leeuw (1979). The data are taken from the *Power in the Classroom* study of Gold (1956). Children in elementary school, divided into eight groups, judge how important 17 characteristics of their classmates are for their power in the classroom. Data are the times each of properties were rated as "very important" by the eight age-sex groups. The groups have three-digit codes in the joint configuration plot. The

first digit indicates the school grade (third vs higher), the second the gender of the respondent, the third indicates the importance of the property rated for males or females.

We first give the configuration plot and the Dist-Dhat plot. For the ” f average” analysis.

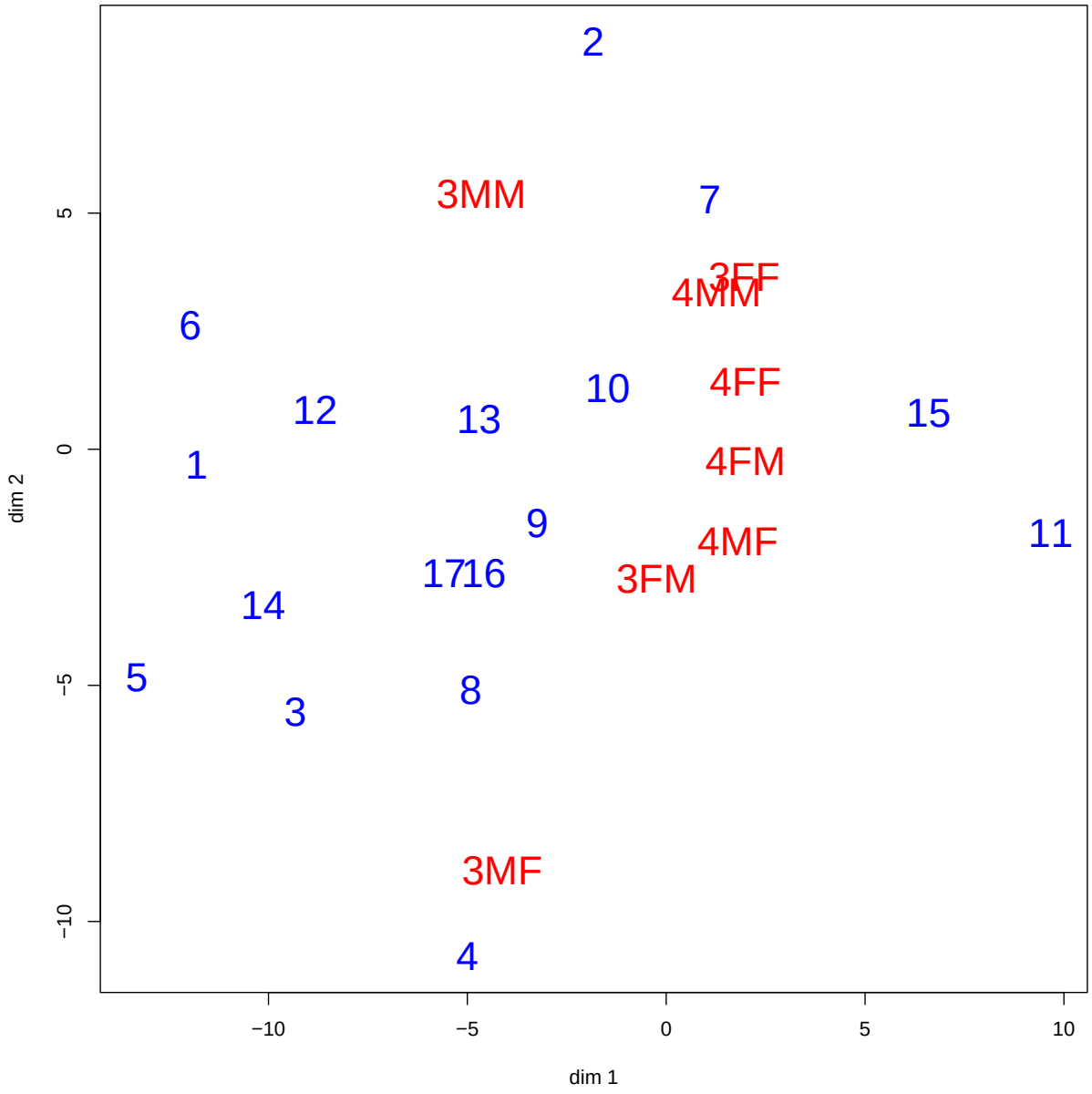

Figure 5: Gold Data, Configuration Plot

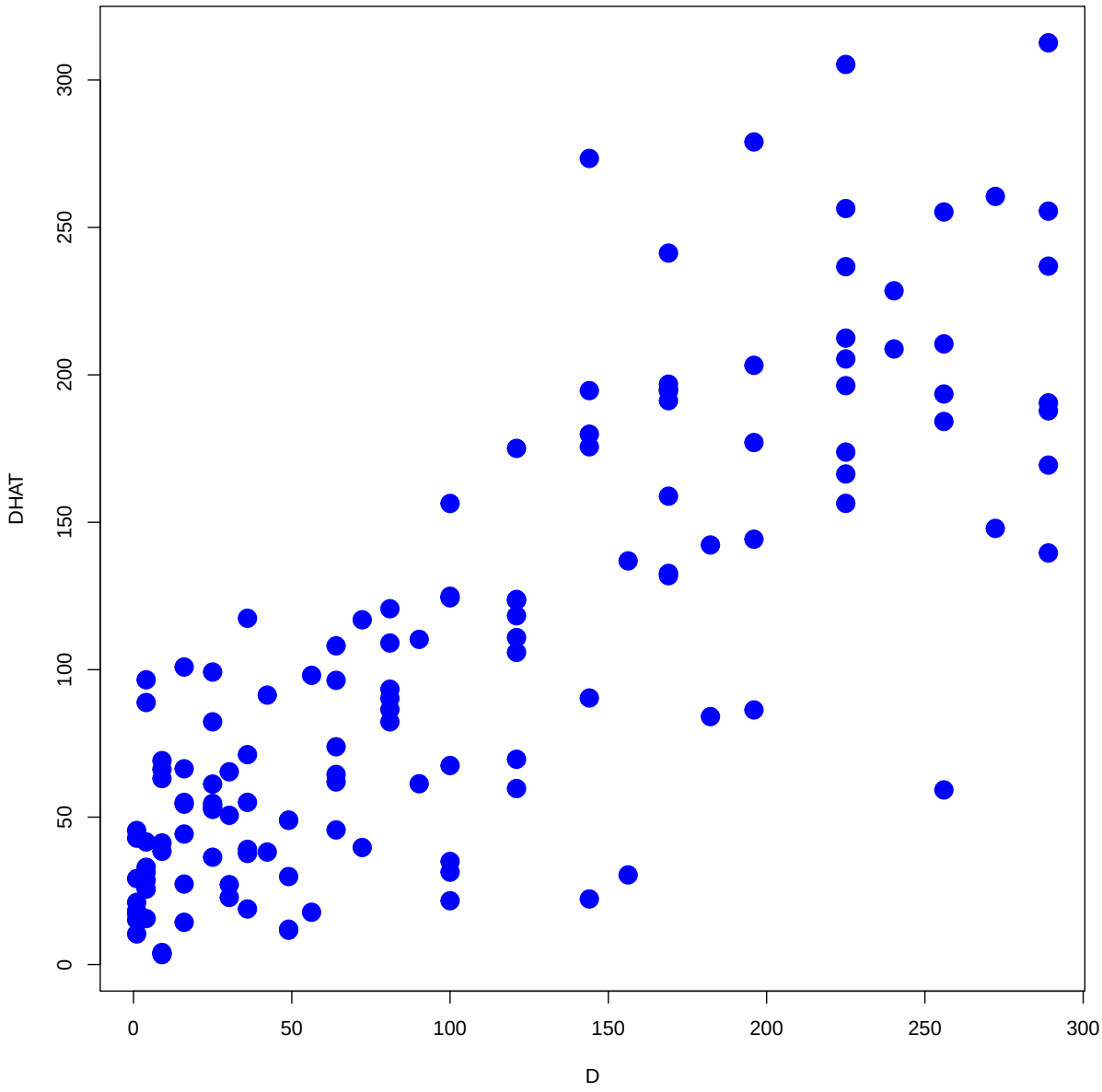

Figure 6: Gold Data, Dist-Dhat Plot

Next, we perform an analysis for each of the columns of F separately, and we plot the solutions for the column scores Y in a single plot (without matching them with rotation and translation). We overlay the result with the result of the average analysis (in red).

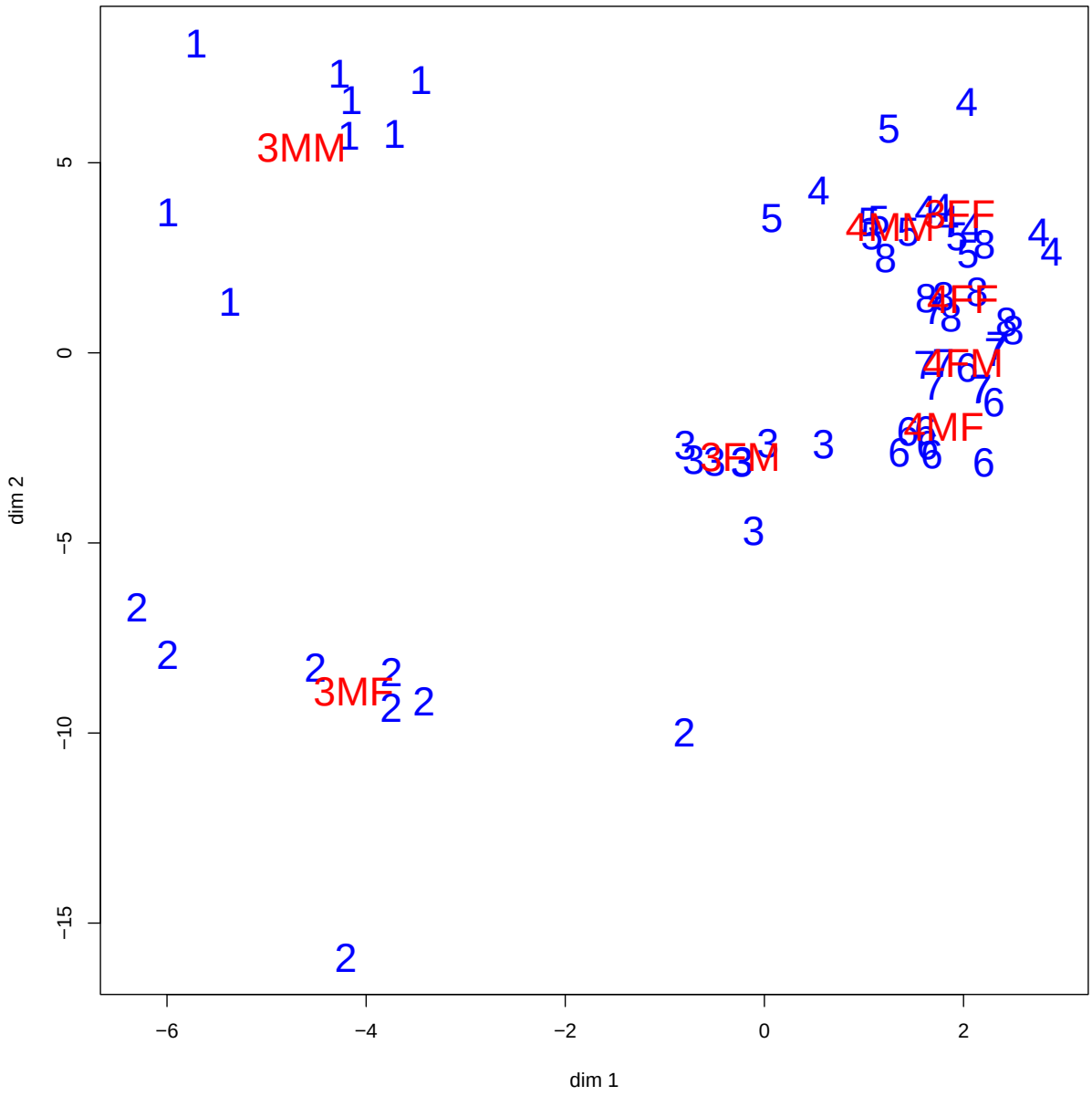

Figure 7: Gold Data, Column Configuration from Eight Column and One Average Analysis

The solution seems to be quite stable over column-selection. It should be noted that if we go to three dimensions, or if we transpose the analysis, the matrix M is no longer positive definite.

5.3 Roskam

In 1968 Roskam (1968) asked 39 of the staff members of the Psychology Laboratory at Nijmegen University to rank the nine different departments in the laboratory as to the importance for their work. The departments are

1. Social Psychology
2. Educational and Developmental Psychology
3. Clinical Psychology
4. Mathematical Psychology and Statistics
5. Experimental Psychology
6. Cultural Psychology and Psychology of Religion
7. Industrial Psychology
8. Test Construction and Validation
9. Physiological and Animal Psychology

Thus the data are 39 rank orders. It seems rather procrustean to approximate discrete rank numbers by continuous distances. Moreover the data are conditional, which means that a rank equal to five for staff member A may not have the same "meaning" as a five for staff member B. Nevertheless we use these data for our example, since it is after all an off-diagonal rectangular matrix with positive numbers.

All rows of D add up to 45, so if we were to choose row-means for f they would be exactly zero. If we use all of the 9 rows of F in turn, then for six of them we find that M is indefinite. The configurations for rows 1, 2, and 7 are plotted. There is clearly a worse fit, and much less stability, than in the Gold example. The most important difference between the three analyses is the relative size of the row and column configurations.

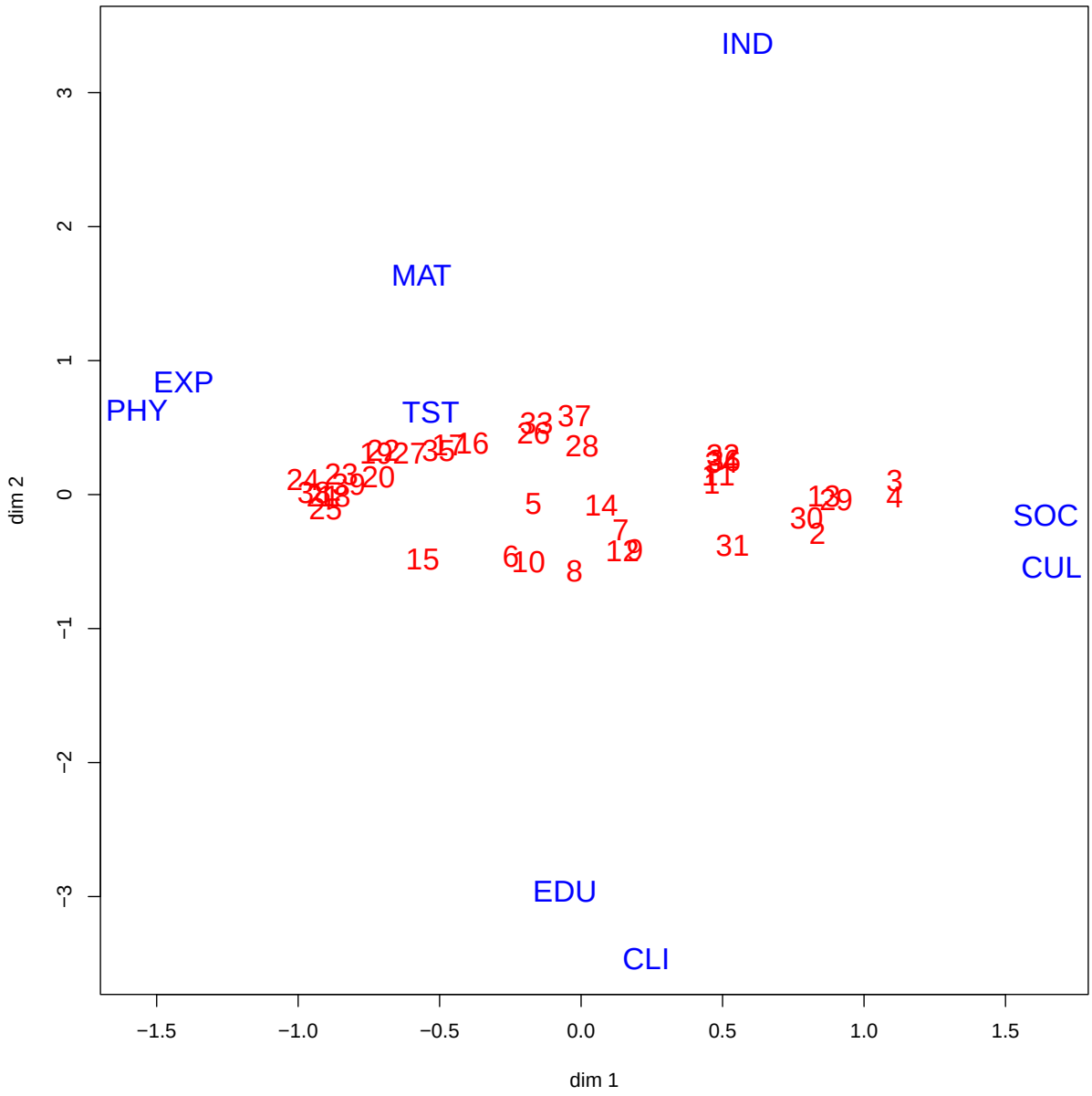

Figure 8: Roskam Data, Configuration Plot from Column 1

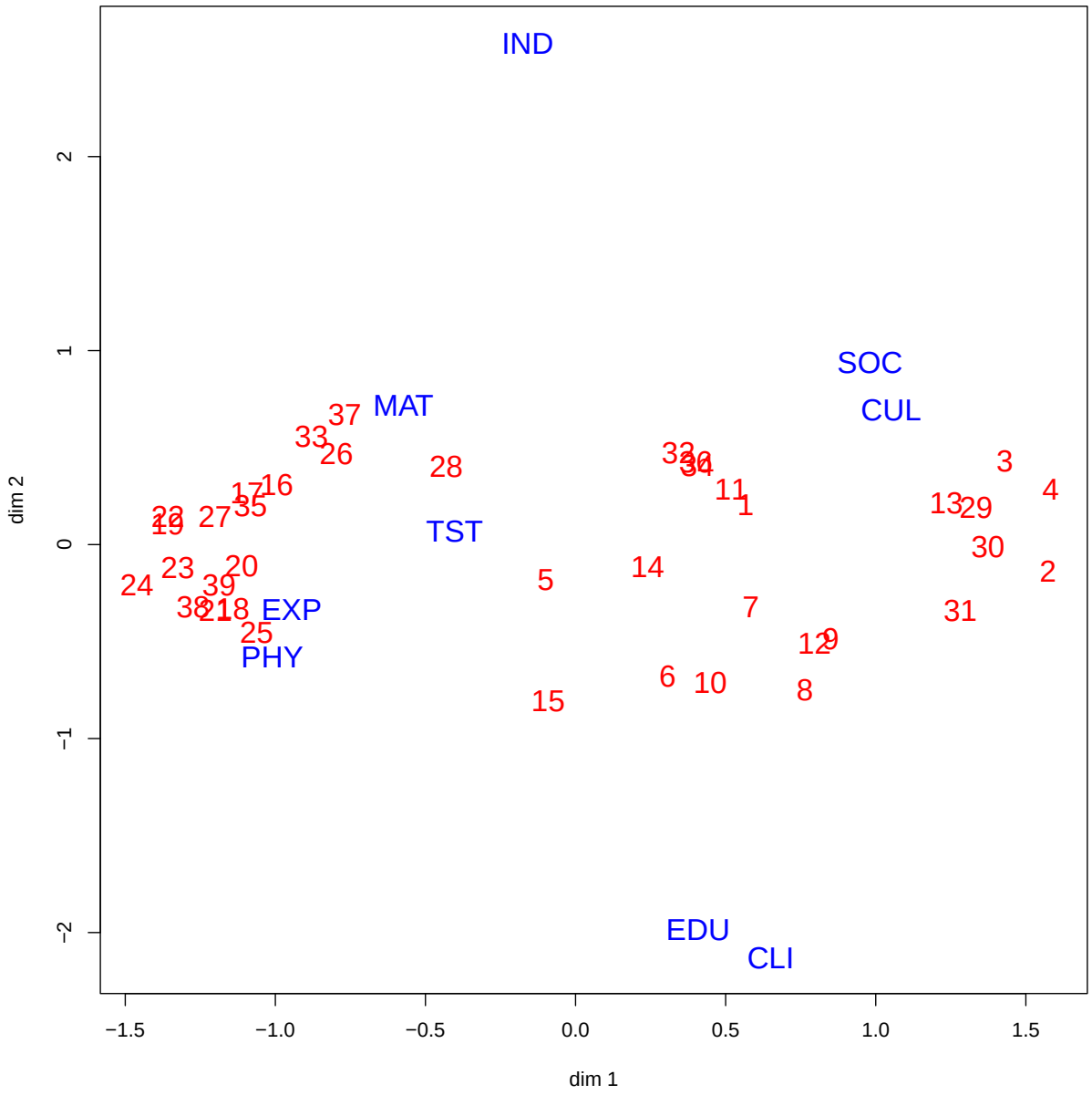

Figure 9: Roskam Data, Configuration Plot from Column 2

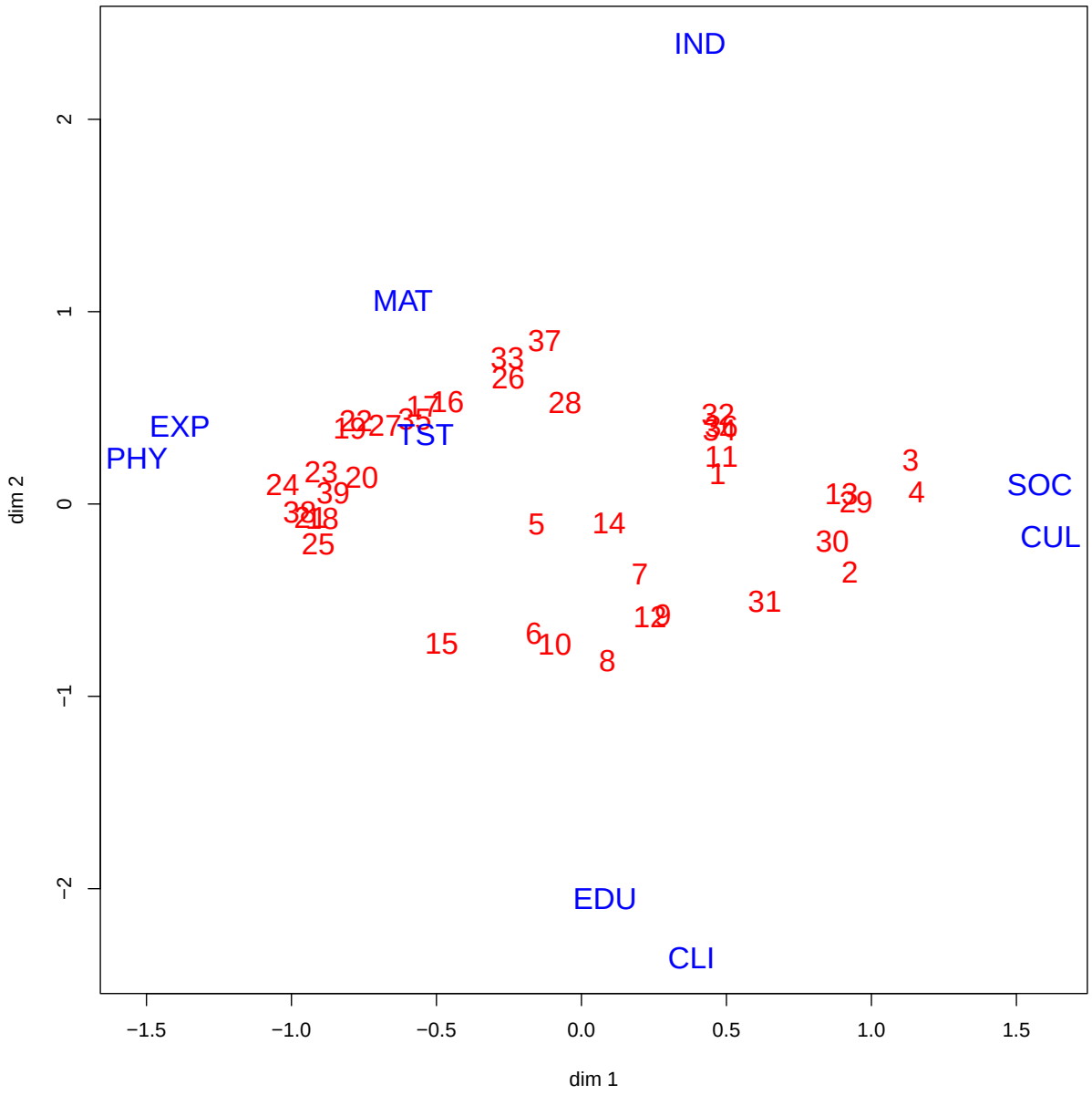

Figure 10: Roskam Data, Configuration Plot from Column 7

The D-Dhat plot for the column seven solution shows the poor fit (due mostly to fitting distances to rank numbers). The red line shows the average Dhat value for each distance category, indicating at least some degree of monotonicity.

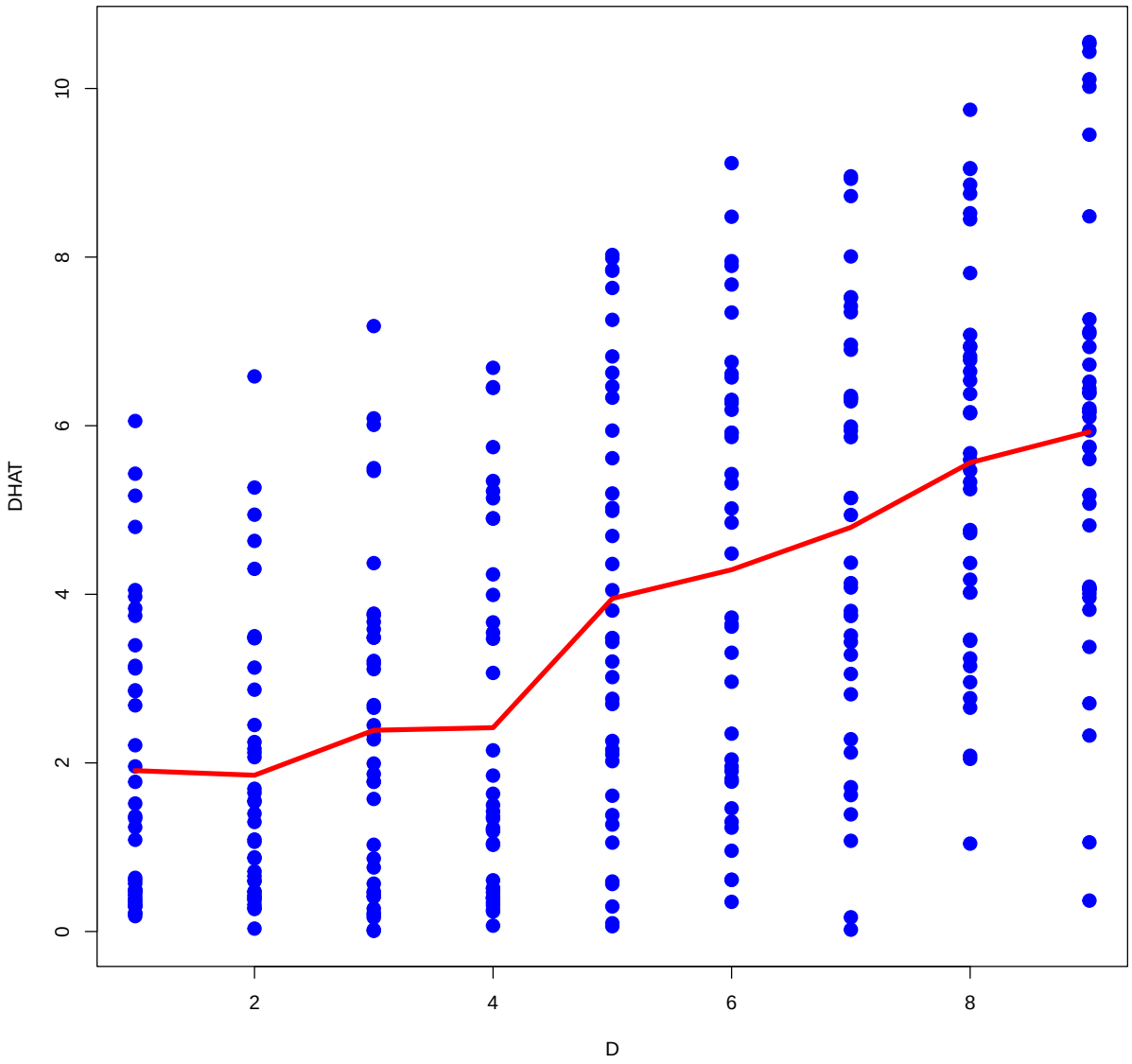

Figure 11: Roskam Data, D-DHAT plot, Column 7 Solution

6 Software

The function schoenemann() in the file schoenemann.R implements the "row-oriented" solution to perfect or imperfect data. We choose a value of p, with $p = 2$ the default. The singular value decomposition is used to approximate D by a matrix of rank p. We then compute $\tilde{F}$. If the parameter fsel is zero, f are the row-averages of $\tilde{F}$. If fsel is equal to j then f is column j of $\tilde{F}$. The R function lm.fit() is used to find the least squares solution of the linear system ..., and the program gives a warning if the estimated rank of K is less than its number of columns $\frac{1}{2}p(p+3)$. From the solution we reconstruct M, and again a warning is issued if M is not positive definite. The R function eigen() uses $M = L\Lambda L'$ to test for positive definiteness. If some eigenvalues are negative then we replace them by their absolute values. There is not real convincing justification for this, except the lame excuse that if M, which is of order p, only has small negative eigenvalues then we only make a small unjustified adjustment. The function schoenemann() does return $\tilde{F}$, K, and M so we can do some post-hoc analyses to see where things went wrong. In addition solutions X and Y, with corresponding $D(X, Y)$ are returned, as well as the input matrix D.

References

Borg, Ingwer, and Patrick J. F. Groenen. 2005. *Modern Multidimensional Scaling*. Second Edition. Springer.

Gold, Martin. 1956. "Power in the Classroom." *Sociometry* 21 (1): 50–60.

Gower, John C. 1966. "Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis." *Biometrika* 53: 325–38.

Heiser, Willem J., and Jan De Leeuw. 1979. "Metric Multidimensional Unfolding." *Methoden En Data Nieuwsbrief SWS/VVS* 4: 26–50. https://jansweb.netlify.app/publication/heiser-deleeuw-a-79/heiser-deleeuw-a-79.pdf.

Roskam, E. E. 1968. "Metric Analysis of Ordinal Data in Psychology." PhD thesis, University of Leiden.

Schoenberg, Isaac J. 1935. "Remarks to Maurice Frechet's article: Sur la Definition Axiomatique d'une Classe d'Espaces Vectoriels Distancies Applicables Vectoriellement sur l'Espace de Hllbert." *Annals of Mathematics* 36: 724–32.

Schönemann, Peter H. 1970. "On Metric Multidimensional Unfolding." *Psychometrika* 35 (3): 349–66.

Torgerson, Warren S. 1958. *Theory and Methods of Scaling*. New York: Wiley.

Young, Gale, and Alston S. Householder. 1938. "Discussion of a Set of Points in Terms of Their Mutual Distances." *Psychometrika* 3 (19-22).