wAddPCA

Jan de Leeuw - University of California Los Angeles
Jan Graffelman - Universitat Politècnica de Catalunya

Started October 06 2022, Version of October 10, 2022

Abstract

TBA

Contents

```
source("wPCA.R")
source("HeartAttack.R")
```

1 General

1.1 Data and Weight Matrices

In this paper we approximate in the weighted least squares sense an $n \times m$ data matrix X by a low-rank approximation matrix Y, which depends on a number of parameters. The weights are collected in an $n \times m$ non-negative matrx W of weights.

For the data X we distinguish two cases:

1. Symmetric Case: X and W are symmetric.
2. Rectangular Case: X and W are rectangular.

And for the weights W we distinguish:

1. Unit Weights: $W = ee'$,
2. Rank One Weights: $W = uv'$,
3. General case: any non-negative W.

Here e is a vector with all elements equal to one.

1.2 Loss Function and Parameters

We minimize a loss function of the form

$$\sigma(Y) := \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}(x_{ij} - y_{ij})^2, \tag{1}$$

with

$$y_{ij} = \delta + p_i + q_j + \sum_{s=1}^{r} a_{is}b_{js}. \tag{2}$$

Thus minimization is over the scalar δ, over the vectors p and q, and over the matrices A and B.

1.2.1 Parameter Restrictions

We distinguish some special cases of equation (2).

1. The null case: $\delta = 0$, $p = 0$, $q = 0$ and this $\delta_{ij} = 0$.
2. The constant case: $p = 0$, $q = 0$, and thus $\delta_{ij} = \delta$.
3. The row case: $q = 0$ and thus $\delta_{ij} = \delta + p_i$.
4. The column case: $p = 0$ and thus $\delta_{ij} = \delta + q_j'$.
5. The complete case: $\delta_{ij} = \delta + p_i + q_j$.

1.2.2 Matrix Form

We can write equation (2) as

$$Y = \delta ee' + pe' + eq' + AB', \tag{3}$$

where the vectors e have all elements equal to one (and are of the appropriate length).

Even more compactly we can define the partitioned matrices

$$\tilde{A} := \begin{bmatrix}\delta e & \mid & p & \mid & e & \mid & A\end{bmatrix}, \tag{4}$$

$$\tilde{B} := \begin{bmatrix}e & \mid & e & \mid & q & \mid & B\end{bmatrix}, \tag{5}$$

and write equation (2) simply as

$$Y = \tilde{A}\tilde{B}'. \tag{6}$$

This shows our technique is a form on weighted and constrained PCA (Takane (2014)), with the constraints given by equations (4) and (5). The matrices $\tilde{A}$ and $\tilde{B}$ have $r + 3$ columns, although clearly their rank is less than or equal to $r + 2$.

1.2.3 Identification

The decomposition (3) is not uniquely defined. There is, of course, the indeterminacy in $Z = AB'$, which defines A and B up to a non-singular linear transformation. We will accept that sad fact as unvoidable and concentrate on the lack of addifiability in the additive decomposition into the four components of equation (3).

Suppose X and W are any two $n \times m$ matrices, with W non-negative and not completely zero. Without loss of generality we can write $X = AB'$ for some $n \times r$ matrix A and some $m \times r$ matrix B with $r \geq \text{rank}(X)$. Define the marginals

$$t := e'We, \tag{7}$$
$$u := We/t \tag{8}$$
$$v := W'e/t, \tag{9}$$

the weighted averages

$$x_{\bullet i} = x_i'u, \tag{10}$$
$$x_{\bullet j} = x_j'v, \tag{11}$$

and the projectors

$$Q_1 = eu', \tag{12}$$
$$Q_2 = ev', \tag{13}$$
$$P_1 = I - eu', \tag{14}$$
$$P_2 = I - ev'. \tag{15}$$

Here P_1 and P_2 transform any vector to deviations from the weighted mean (weighted by u and v), while Q_1 and Q_2 replace the elements of any vector by their weighted mean. Thus $P_1e = 0, P_2e = 0, Q_1e = e, Q_2e = 2$ and $u'P_1 = 0, u'P_2 = 0, u'Q_1 = u', u'Q_2 = u'$.

Now obviously

$$X = (P_1 + Q_1)X(P_2 + Q_2)' = P_1XP_2' + P_1XQ_2' + Q_1XP_2' + Q_1XQ_2', \tag{16}$$

which is in elementwise notation

$$x_{ij} = (x_{ij} - x_{i\bullet} - x_{\bullet j} + x_{\bullet\bullet}) + (x_{i\bullet} - x_{\bullet\bullet}) + (x_{j\bullet} - x_{\bullet\bullet}) + x_{\bullet\bullet} \tag{17}$$

We call this the canonical additive form of X with respect to W. It is unique, and consequently it solves the additive identification problem.

Now suppose

$$X = \delta ee' + pe' + eq' + AB' \tag{18}$$

What is the canonical additive form in thus case ? Some computation shows

$$P_1XP_2' = (A - ea_\bullet')(B - eb_\bullet')', \tag{19}$$
$$Q_1XP_2' = e((q - q_\bullet)' + a_\bullet'(B - eb_\bullet')'), \tag{20}$$
$$P_1XQ_2' = ((p - p_\bullet) + (A - ea_\bullet')b_\bullet)e', \tag{21}$$
$$Q_1XQ_2' = (\delta + p_\bullet + q_\bullet + a_\bullet'b_\bullet)ee'. \tag{22}$$

This is obviously of the form $\tilde{\delta} + \tilde{p}e' + e\tilde{q}' + \tilde{A}\tilde{B}'$, where the new components are related to the old ones by the formulas (19)-(22).

One problem, however. In the column case with $p = 0$ in equation (18) it follows from (21) that we do not necessarily have $\tilde{p} = (A - ea_\bullet')b_\bullet$ equal to zero, and thus the parameter restrictions are not preserved by the canonical form. Same for the row case, of course.The solution is to use $X = (P_1 + Q_1)X$, which gives

$$P_1X = (p - p_\bullet)e' + (A - ea_\bullet')B', \tag{23}$$
$$Q_1X = \delta ee' + p_\bullet e' + eq' + ea_\bullet' B', \tag{24}$$

If $p = 0$

$$P_1X = (A - ea_\bullet')B', \tag{25}$$
$$Q_1X = \delta ee' + e(q + B'a_\bullet)'. \tag{26}$$

This is of the form $\tilde{\delta} + e\tilde{q}' + \tilde{A}B'$, and consequently the parameter restrictionsare preserved.

1.3 Applications of wAddPCA

Special cases are

1. Principal Component Analysis.
2. Simple and Multiple Correspondence Analysis.
3. Joint Correspondence Analysis (Greenacre (1988), McDonald (1969b))
4. FANOVA (Gollob (1968), De Leeuw (1973)).
5. MINRES Factor Analysis (Harman and Jones (1966)).
6. McDonald's Group Factor Analysis (McDonald (1969a))
7. Bailey-Gower Weighted Fitting (Bailey and Gower (1990))

All these special cases can be with or without missing data. We will give examples later on in this paper or in subsequent papers.

2 Rectangular Case

We use the Heart Attack data from Saporta (2006), section 18.5.2, previously usedby Graffelman (1922). The data frame has 101 observations on 7 variables.

```
x <- as.matrix(HeartAttack[,1:7])
m <- 1 / apply(x, 2, max)
x <- x %*% diag(m)
```

Thus we normalize the data matrix (in which all entries are positive) to the unit interval by dividing each column by its maximum value.

2.1 Unit Weights, a.k.a. Unweighted PCA

Set all weights equal to one.

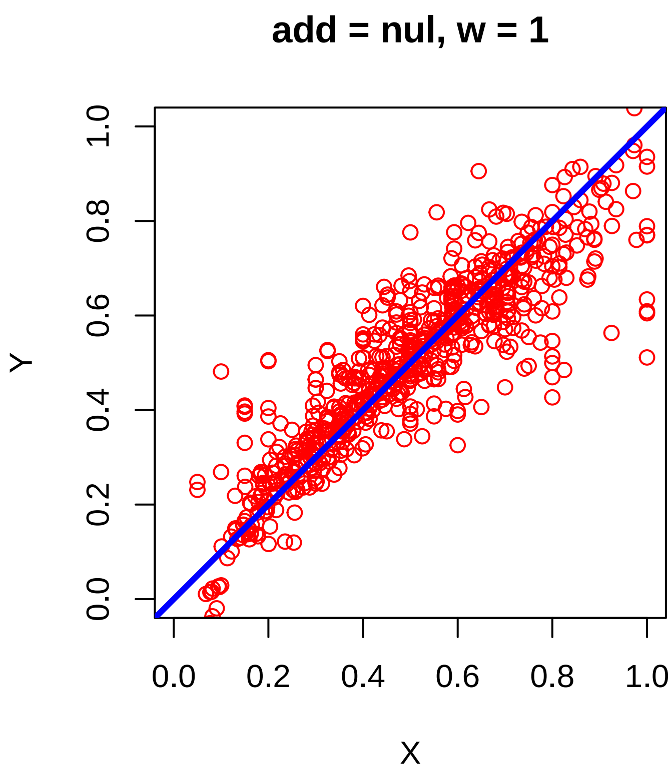
```
w1r <- matrix(1, 101, 7)
```

2.1.1 add = "nul"

If add = "nul" we actually compute the p largest singular values and vectors of X, i.e. we do an unweighted principal component analysis. For add = "nul" there are no outer iterations.

```
znul <- wAddPCA(x, w1r, add = "nul", verboseout = FALSE)
```

The minimum loss function value is 6.645561. We plot X versus $Y = AB'$.

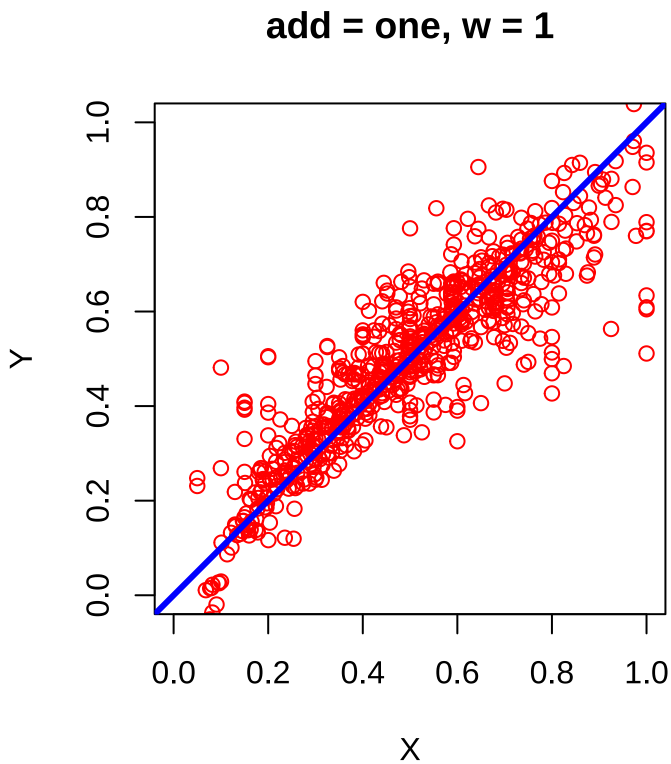


2.1.2 add = "one"

This is the case studied, and advocated, by Graffelman (1922).

```
zone <- wAddPCA(x, w1r, add = "one", itmaxout = 10000, verboseout = FALSE)
```

The minimum loss function value after 1513 outer iterations is 6.619003. We plot X versus $Y = \delta ee' + AB'$, where δ is 0.154970.

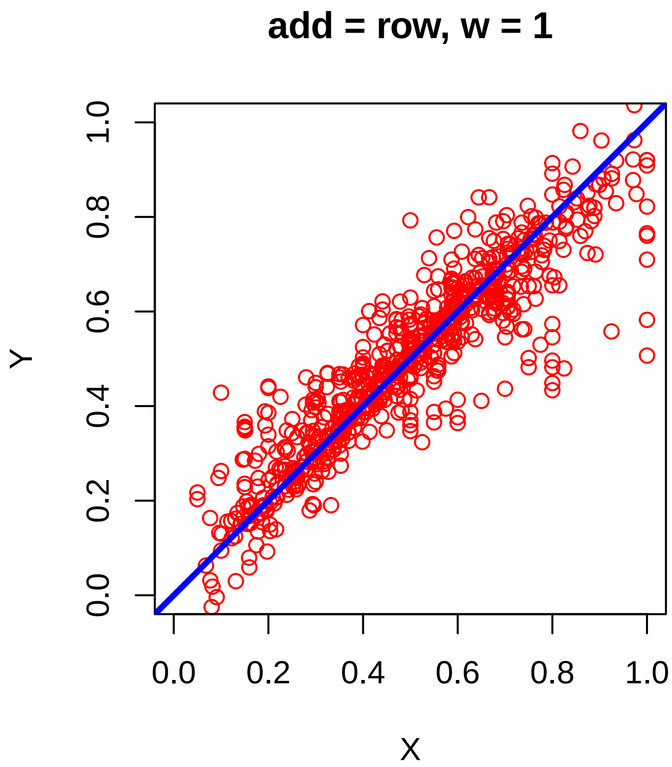


2.1.3 add = "row"

This analysis is equivalent to a PCA after subtracting the mean from each row of X, which is consequently related to the Q-technique of Stephenson (1953).

```
zrow <- wAddPCA(x, w1r, add = "row", verboseout = FALSE)
```

The minimum loss function value after 104 outer iterations is 5.508861. We plot X versus $Y = pe' + AB'$.

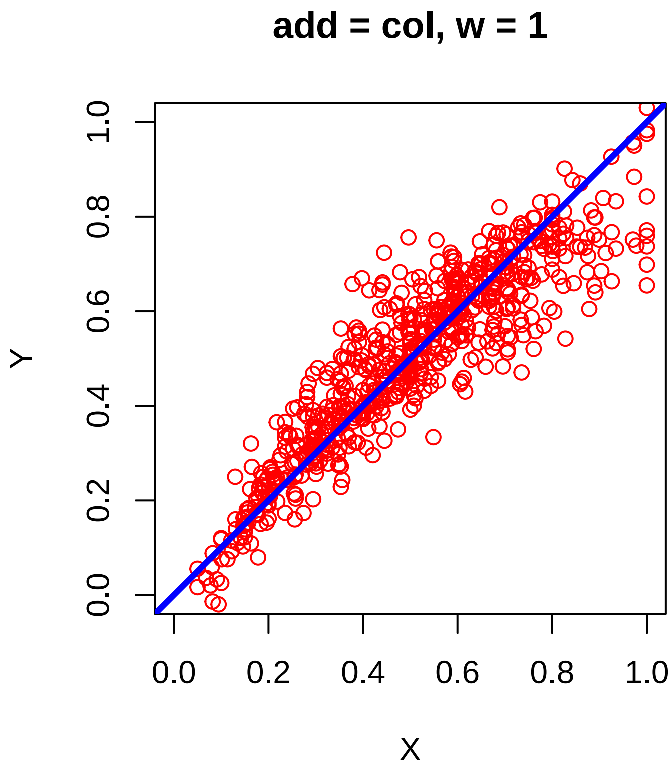


2.1.4 add = "col"

This is the more familiar technique of subtracting the column means before doing the PCA.

```
zcol <- wAddPCA(x, w1r, add = "col", itmaxout = 10000, verboseout = FALSE)
```

The minimum loss function value after 1531 outer iterations is 5.475089. We plot X versus $Y = eq' + AB'$.

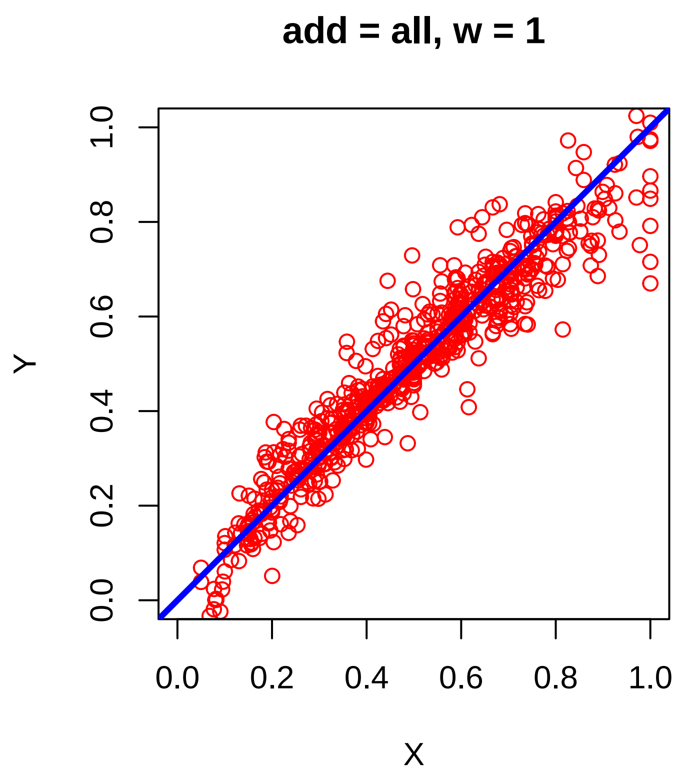


2.1.5 add = "all"

This is the FANOVA of Gollob (1968).

```
zall <- wAddPCA(x, w1r, add = "all", itmaxout = 10000, verboseout = FALSE)
```

The minimum loss function value after 112 outer iterations is 2.784428. We plot X versus



$Y = pe' + eq' + AB'$.

2.2 Zero-One Weights, a.k.a. Missing Data PCA

We redefine W by setting $w_{ij} = 1$ if $i + j$ is odd and $w_{ij} = 0$ if $i + j$ is even.

```
wmr <- outer(1:101, 1:7, "+") %% 2
```

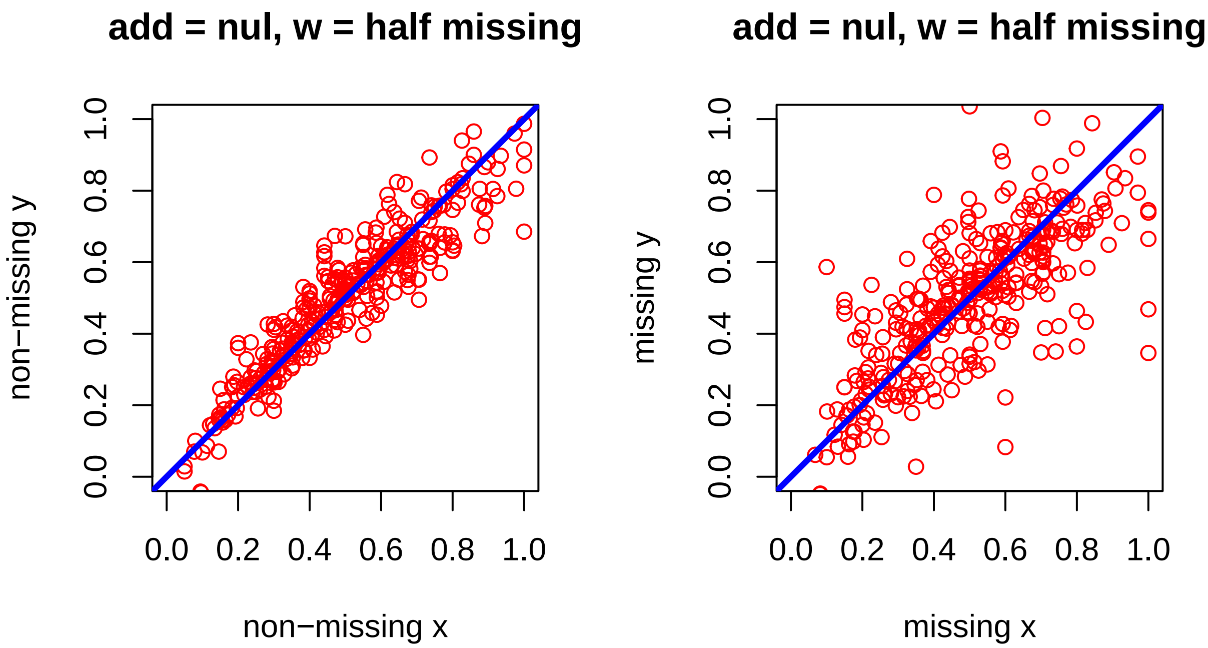
This creates a 101×7 matrix with 50.0707 percent missing. In this case our loss function measures the discrepancy between the non-missing observed data in X and the corresponding elements of Z. Because the algorithm generates all of Z and we know the elements of X that got zero weight, and thus did not influence the results of the analysis, we can actually also compute the discrepancy between the ibserve and reconstructed "missing" elememnts. This is a first step twowards cross-validation.

2.2.1 add="nul"

This is a PCA of an incomplete data matrix, which can be considered a low-rank matrix completion technique.

```
xnul <- wAddPCA(x, wmr, add = "nul", verboseout = FALSE)
```

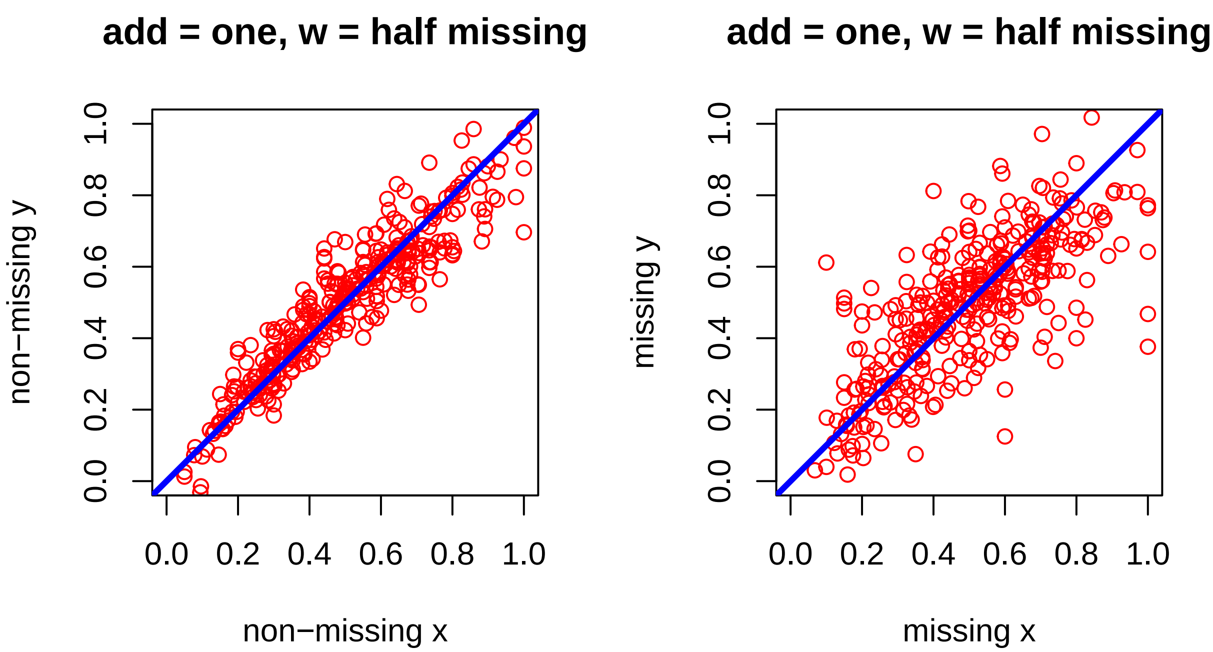
The minimum loss function value after 2 iterations is 2.075267, which "explains" 97.9819 percent of the sum of squares of the non-missing elements of X. The "cross-validated" sum of squares is 7.480425, which "explains" 92.8191 percent of the sum of squares of the elements of X we made missing.



2.2.2 add="one"

```
xone <- wAddPCA(x, wmr, add = "one", verboseout = FALSE, itmaxout = 10000)
```

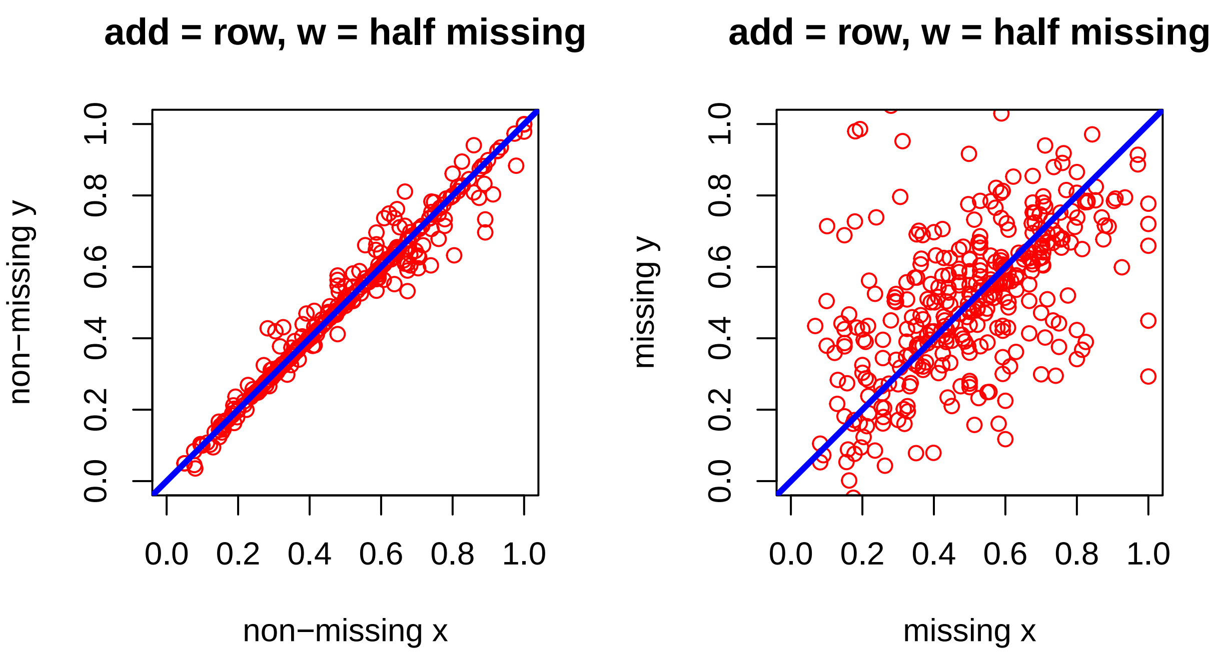
The minimum loss function value after 4350 iterations is 2.062519, which "explains" 97.9943 percent of the sum of squares of the non-missing elements of X. The cross-validated sum of squares is 7.715528, which "explains" 92.5934 percent of the sum of squares of the elements of X we made missing. Note that this is worse than add="nul", despite the extra parameter δ, which is -0.2217.



2.2.3 add="row"

```
xrow <- wAddPCA(x, wmr, add = "row", verboseout = FALSE)
```

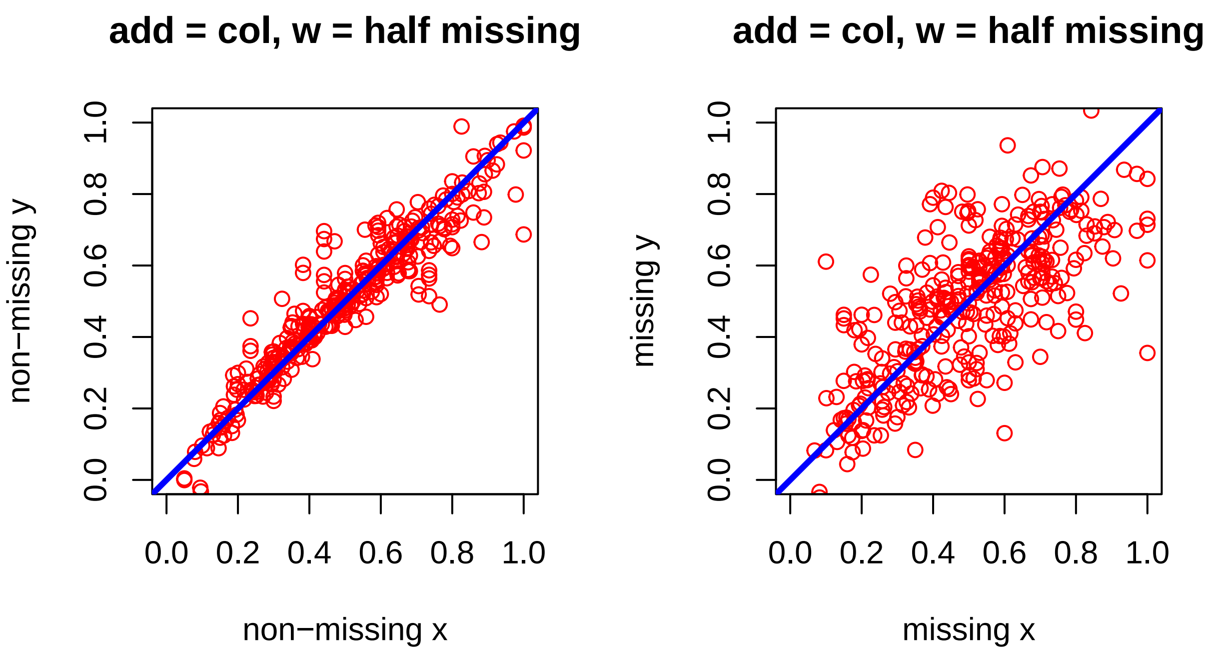
The minimum loss function value after 210 iterations is 0.512344, which "explains" 99.5018 percent of the sum of squares of the non-missing elements of X. The cross-validated sum of squares is 14.862530, which "explains" 85.7325 percent of the sum of squares of the elements of X we made missing. Adding more parameters makes the cross-validated sum of squares worse, at least in this example.



2.2.4 add="col"

```r
xcol <- wAddPCA(x, wmr, add = "col", verboseout = FALSE)
```

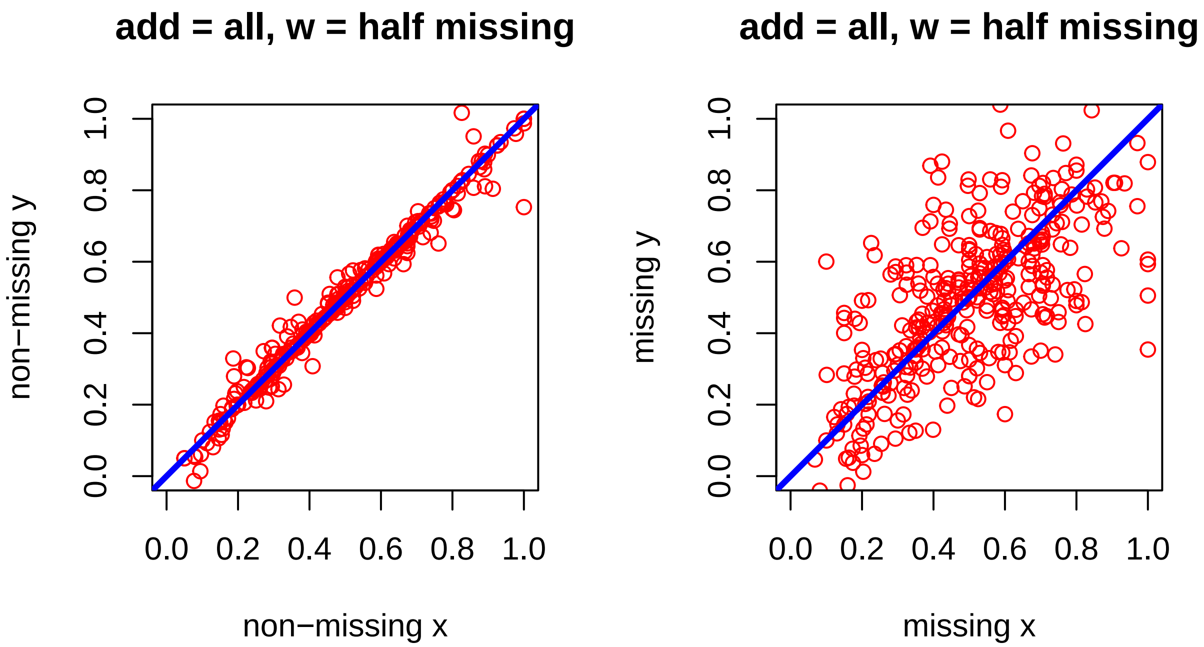
The minimum loss function value after 516 iterations is 1.653098, which "explains" 98.3925 percent of the sum of squares of the non-missing elements of X. The cross-validated sum of squares is 8.953942, which "explains" 91.4046 percent of the sum of squares of the elements of X we made missing.



2.2.5 add="all"

```r
xall <- wAddPCA(x, wmr, add = "all", verboseout = FALSE)
```

The minimum loss function value after 331 iterations is 0.375508, which "explains" 99.6348 percent of the sum of squares of the non-missing elements of X. The cross-validated sum of squares is 10.602715, which "explains" 89.8218 percent of the sum of squares of the elements of X we made missing.



3 Symmetric Case

3.1 Unit Weights, PCA of Correlations

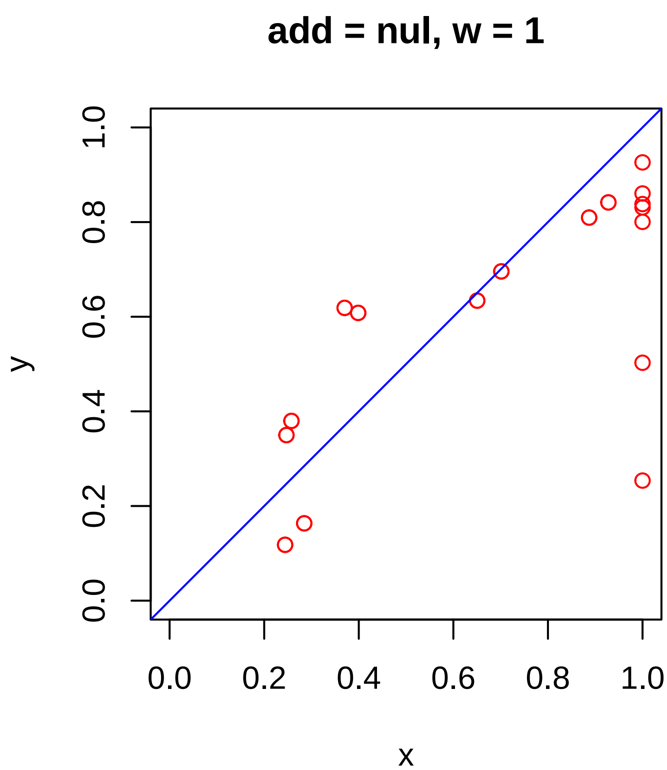
We continue to use the HeartAttack data, but this time the correlation matrix of the seven variables. Because of symmetry the add="row" and add="col" options do not make much sense, and we limit ourselves to add="nul", add ="one", and add = "all".

```
r <- cor(x)
w1s <- matrix(1, 7, 7)
```

3.1.1 add = "nul"

```
rnul <- wAddPCA(r, w1s, add = "nul", verboseout = FALSE)
```

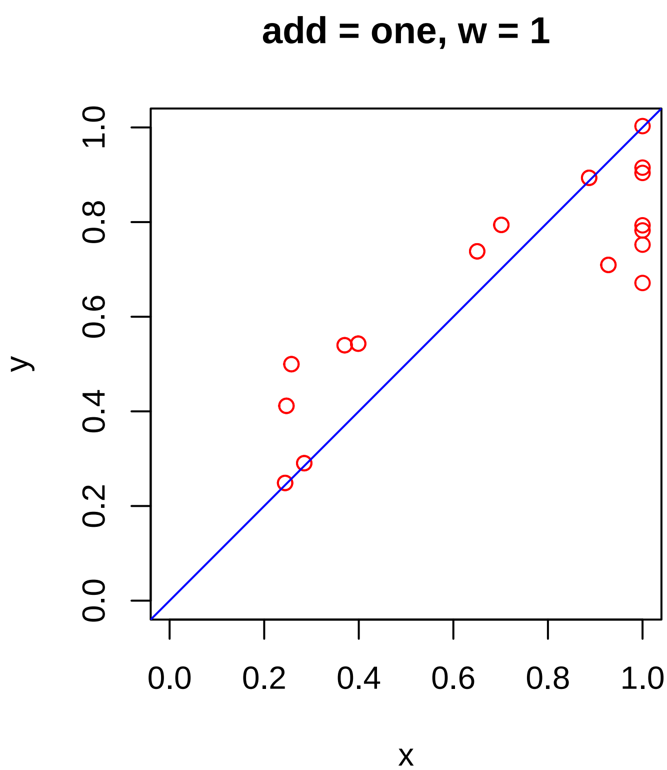
The minimum loss function value after 1 iterations is 1.654913, which "explains" 90.5412 percent of the sum of squares of the elements of R. Note that the diagonal elements of R are "explained" badly.



3.1.2 add = "one"

```
rone <- wAddPCA(r, w1s, add = "one", verboseout = FALSE)
```

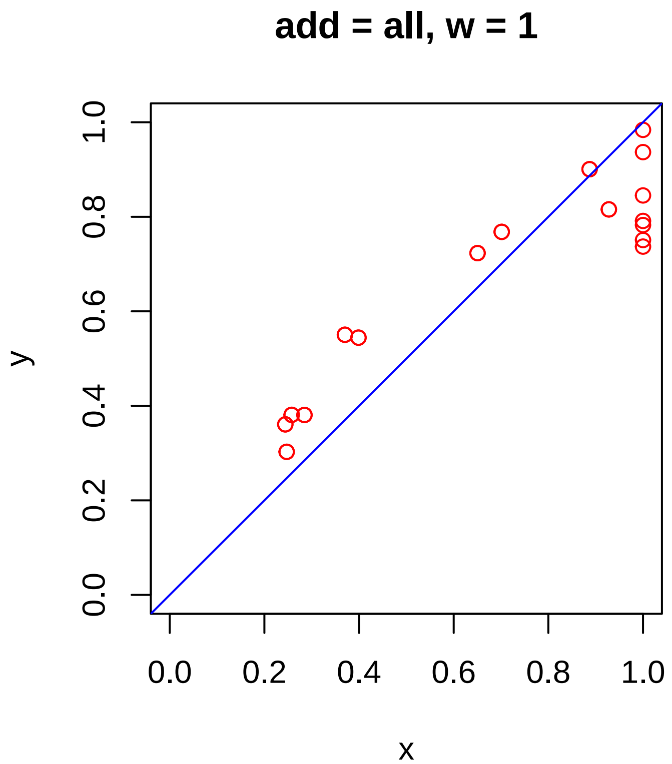
The minimum loss function value after 7 iterations is 1.034593, which "explains" 94.0867 percent of the sum of squares of the elements of R. For δ we find 0.137361.



3.1.3 add = "all"

```
rall <- wAddPCA(r, w1s, add = "all", verboseout = FALSE)
```

The minimum loss function value after 10 iterations is 0.642334, which "explains" 96.3287 percent of the sum of squares of the elements of R.



3.2 Off-diagonal Weights, MINRES Factor Analysis

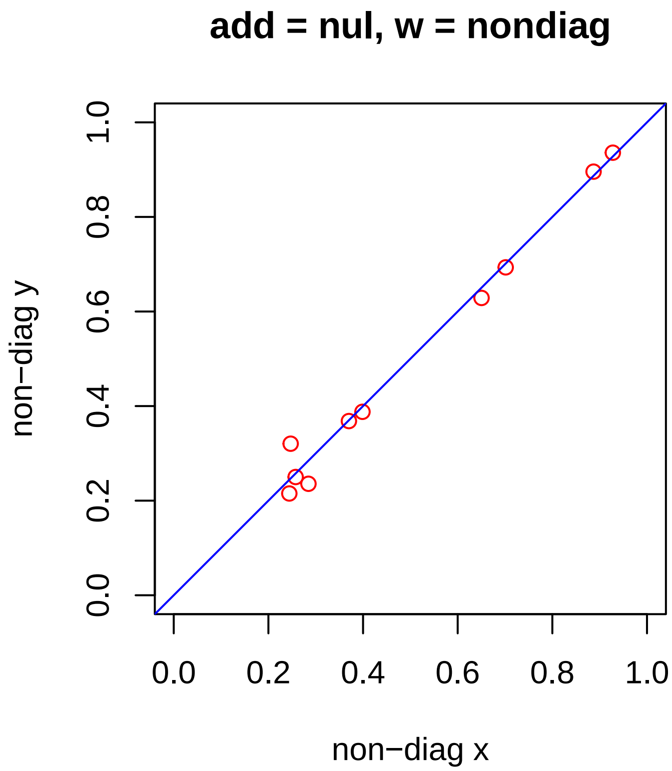
MINRES factor analysis (Harman and Jones (1966)) treats the diagonal of the correlation or covariance matrix as missing. It seems rather nonsensical in that case to think in terms of cross-validation because we are not reconstructing missing elements, but we are computing communality estimates.

```
wms <- 1 - diag(7)
```

3.2.1 add = "nul"

```
snul <- wAddPCA(r, wms, add = "nul", verboseout = FALSE)
```

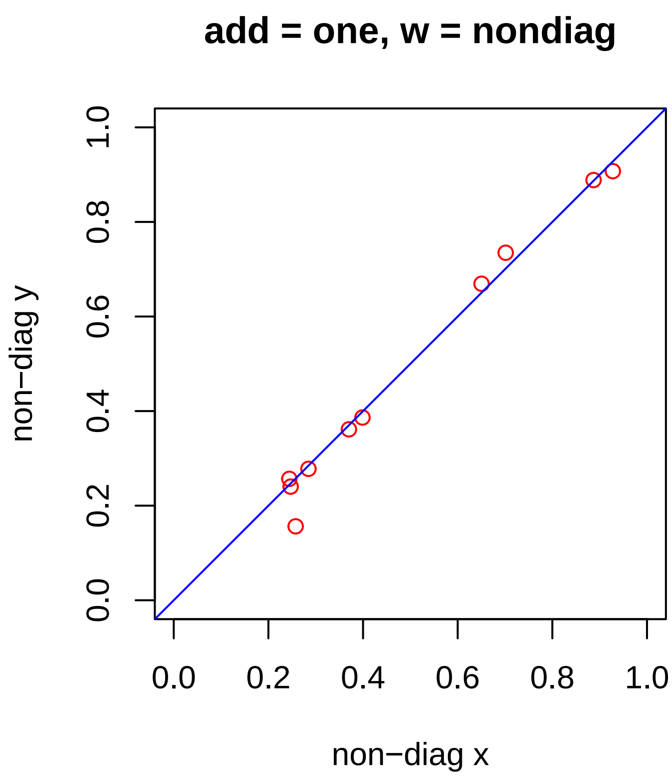
The minimum loss function value after 2 iterations is 0.240373, which "explains" 97.7099 percent of the sum of squares of the non-diagonal elements of R. The estimated communalities are 0.162227, 1.060222, 0.815654, 0.968165, 0.911064, 0.082072, 0.763548. Note the Heywood case.



3.2.2 add = "one"

```
sone <- wAddPCA(r, wms, add = "one", verboseout = FALSE)
```

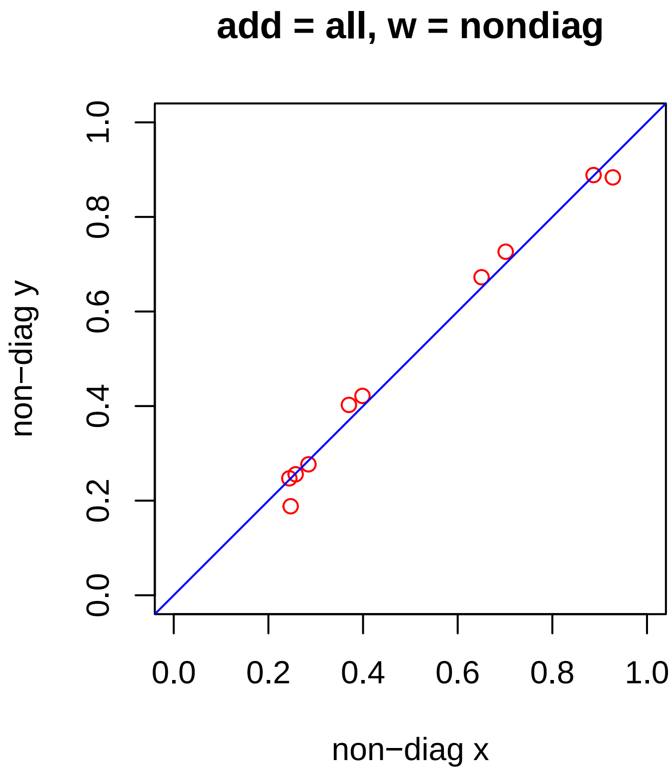
The minimum loss function value after 356 iterations is 0.166553, which "explains" 98.4132 percent of the sum of squares of the non-diagonal elements of R. For δ we find -0.289836. The estimated communalities are 0.080815, 1.678535, 0.440120, 0.962580, 0.858920, -0.029041, 0.646552. We always include the δ_{ii} in the communality estimate.



3.2.3 add = "all"

```
sall <- wAddPCA(r, wms, add = "all", verboseout = FALSE)
```

The minimum loss function value after 453 iterations is 0.018130, which "explains" 99.8273 percent of the sum of squares of the non-diagonal elements of R. The estimated communalities are 0.048063, 2.624376, -0.078584, 0.929286, 0.843411, 1.211455, 0.563824. If we do not include the δ_{ii}, but just the $a_i'a_i$, they are 0.432163, 2.255928, 0.394067, 0.826656, 0.740873, 0.492276, 0.784958.



References

Bailey, R. A., and J. C. Gower. 1990. "Approximating a Symmetric Matrix." *Psychometrika* 55 (4): 665–75.

De Leeuw, J. 1973. "A Generalization of the Young-Whittle Model." Research Report 006-73. Leiden, The Netherlands: Department of Data Theory FSW/RUL.

Gollob, H. F. 1968. "A Statistical Model Which Combines Features of Factor Analytic and Analysis of Variance Techniques." *Psychometrika* 33 (1): 73–115.

Graffelman, J. 1922. "Improved Approximation and Visualization of the Correlation Matrix." *Submitted*.

Greenacre, M. J. 1988. "Correspondence Analysis of Multivariate Categorical Data by Weighted Least-Squares." *Biometrika* 75 (3): 457–67.

Harman, H. H., and W. H. Jones. 1966. "Factor Analysis by Minimizing Residuals ." *Psychometrika* 31: 351–68.

McDonald, R. P. 1969a. "A Generalized Common Factor Analysis Based on Residual Covariance Matrices of Prescribed Structure." *British Journal of Mathematical and Statistical Psychology* 22: 149–63.

———. 1969b. "The Common Factor Analysis of Multicategory Data." *British Journal of Mathematical and Statistical Psychology* 22: 165–75.

Saporta, G. 2006. *Probabilités, Analyse des Données, et Statistiques*. 2nd ed. Éditions Technip.

Stephenson, W. 1953. *The Study of Behavior: Q-technique and its Methodology*. University of Chicago Press.

Takane, Y. 2014. *Constrained Principal Component Analysis and Related Techniques.* Monographs on Statistics and Applied Probability 129. CRC Press.