

## GPArotation Functions

In R, the functions in this package are made available with

```
> library("GPArotation")
```

The most complete reference for the software is: Bernaards, C.A. and Jennrich, R.I. (2005) Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis. Educational and Psychological Measurement. A mirror of the original repository that is referenced in the paper, with additional material is available here:

[https://optimizer.r-forge.r-project.org/GPArotation\\_www/indexOriginal.html](https://optimizer.r-forge.r-project.org/GPArotation_www/indexOriginal.html).

Rotations can be performed by providing an orthogonal matrix to the gradient projection function. Orthogonal matrix for rotation can be obtained by extracting an unrotated factor loadings matrix. A rotation is done by calling the rotation name directly, or by calling one of the wrapper functions `GPFRSorth` or `GPFRSoblq`, for orthogonal and oblique rotation, respectively.

Under the hood, rotations are computed using the Gradient Projection Algorithm code, which can be called directly. The key functionality of the algorithm is included in the `GPForth` and `GPFoblq` functions for orthogonal and oblique rotation, respectively. Calling these functions directly works as it always has (the codes have not changed).

The rotated loadings matrix is the pattern matrix. The structure matrix may be obtained using the `summary` command.

### GPArotation Functions with factanal

The `GPArotation` can be used in conjunction with the built-in R `factanal` function. It is recommended to rotate outside of `factanal`.

```
> data(ability.cov)
> z <- factanal(factors = 2, covmat = ability.cov, rotation = "none")
> # quartimax rotation
> GPFRSorth(loadings(z), method = "quartimax")
> quartimax(z$loadings)
> # oblimin rotation
> GPFRSoblq(z$loadings, method = "oblimin")
> oblimin(loadings(z))
```

**Important note:** `factanal` allows for calling a rotation directly from the `factanal` call. However, due to a `factanal` calculation error in the computation of the correlation matrix, the produced correlation matrix *may* be wrong for oblique rotation (orthogonal rotation are not affected). However, the correlation matrix `Phi` produced by `GPArotation` is the correct correlation matrix when performing rotation outside of `factanal`.

### Recovery of The Unrotated Loadings Matrix

Recovery of the unrotated loadings matrix is consistent with the definitions used in Bernaards & Jennrich (2005) (page 678). For example, the unrotated matrix  $A$  may be recovered as follows.

```

> y <- factanal(factors=3, covmat=ability.cov, rotation = "none")
> y.quart <- quartimax(y$loadings)
> max(abs(y.quart$loadings %*% t(y.quart$Th) - unclass(y$loadings)))
> y.obli <- oblimin(y$loadings, normalize=TRUE, randomStarts=15)
> max(abs(y.obli$loadings %*% t(y.obli$Th) - unclass(y$loadings)))

```

By the same definitions logic, the factor correlation matrix is calculated as Bornaards & Jennrich (2005) (page 695),

```

> y <- factanal(factors=3, covmat=ability.cov, rotation = "none", randomStarts=15)
> y.obli <- oblimin(y$loadings, normalize=TRUE, randomStarts=15)
> max(abs(y.obli$Phi - t(y.obli$Th) %*% y.obli$Th))

```

## Random Starts

If multiple random starts are desired then the `randomStarts` option may be utilized. For example, 100 random starts of the oblique infomax rotation,

```

> data(Thurstone, package = "GPArotation")
> infomaxQ(box26, randomStarts = 100) # 100 random starts
> infomaxQ(box26, Tmat=Random.Start(3)) # a single random start
> infomaxQ(box26, randomStarts = 1) # also a single random start

```

The loadings that are output have the lowest complexity value  $f$ . While the lowest local minimum may be the global minimum solution, technically it can not be guaranteed that the lowest local minimum is in fact the global minimum. To further investigate the local minima it is recommended to use the *fungible* package using the `faMain` function. When in doubt, trying random initial rotation matrix is advised.

For a detailed discussion, consult Nguyen & Waller (2022). Additional algorithmic considerations are in Bornaards & Jennrich (2005) (page 680).

## An Example of Target Rotation

Fischer & Fontaine (2010) describe measuring self-reported extra-role behavior in samples of British and East German employees. They publish rotation matrices for two samples, and investigate the structural equivalence of the loadings matrices. Additional context is available in the manuscript. Structural equivalence includes target rotation, as well as calculation of a number of agreement coefficients. The table lists the varimax rotated loadings matrices. Performing target rotation of one loadings matrix to the other can help in interpreting assessing equivalence.

	Britain		East Germany	
	Factor 1	Factor 2	Factor 1	Factor 2
I am always punctual.	.783	-.163	.778	-.066
I do not take extra breaks.	.811	.202	.875	.081
I follow work rules and instructions with extreme care.	.724	.209	.751	.079
I never take long lunches or breaks.	.850	.064	.739	.092
I search for causes for something . that did not function properly.	-.031	.592	.195	.574
I often motivate others to express their ideas and opinions.	-.028	.723	-.030	.807
During the last year I changed something. in my work.	.388	.434	-.135	.717
I encourage others to speak up at meetings.	.141	.808	.125	.738
I continuously try to submit suggestions to improve my work.	.215	.709	.060	.691

The varimax rotations for each of the samples may be expected to be similar because the two loadings matrices are from different samples measuring the same constructs. Below are target rotation of the East German loadings matrix towards the Britain one, followed by calculation of agreement coefficients. Fischer & Fontaine (2010) note that coefficients generally should be “beyond the commonly accepted value of 0.90.”

```

> origdigits <- options("digits")
> options(digits = 2)
> trBritain <- matrix( c(.783,-.163,.811,.202,.724,.209,.850,.064,
-.031,.592,-.028,.723,.388,.434,.141,.808,.215,.709), byrow=TRUE, ncol=2)
> trGermany <- matrix( c(.778,-.066,.875,.081,.751,.079,.739,.092,
.195,.574,-.030,.807,-.135,.717,.125,.738,.060,.691), byrow=TRUE, ncol = 2)
> # orthogonal rotation of trGermany towards trBritain
> trx <- targetT(trGermany, Target = trBritain)
> # Factor loadings after target rotation
> trx
> # Differences between loadings matrices after rotation
> y <- trx$loadings - trBritain
> print(y, digits = 1)
> # Square Root of the mean squared difference per item
> sqrt(apply((y^2), 1, mean))
> # Square Root of the mean squared difference per factor
> sqrt(apply((y^2), 2, mean))
> # Identity coefficient per factor after rotation
> 2 * colSums(trx$loadings*trBritain)/( colSums(trx$loadings^2)+colSums(trBritain^2))
> # Additivity coefficient per factor after rotation
> diag(2 * cov(trx$loadings, trBritain) ) / diag(var(trx$loadings)+var(trBritain))
> # Proportionality coefficient per factor after rotation
> colSums(trBritain * trx$loadings)/sqrt(colSums(trBritain^2)*colSums(trx$loadings^2))
> # Correlation for each factor per factor after rotation
> diag(cor(trBritain, trx$loadings))

```

```
> options(digits = origdigits$digits)
```

## An Example of Partially Specified Target Rotation

Browne (1972) reported an initial loadings matrix and a partially specified target to rotated towards. In *GPArotation* the partially specified target matrix is of the same dimension as the initial matrix *A*, and with NA in the matrix entries that are not pre-specified. Both procedures target rotation and partially specified target rotation can be used to reproduce Browne (1972) results.

In this orthogonal rotation example, `targetT` includes a `Target` matrix with NA in entries not used in target rotation. With `pst` no missing values are present in the `Target` matrix, and the weight matrix *W* includes weight 0 for entries not used, and 1 for entries included in the rotation.

```
> A <- matrix(c(.664, .688, .492, .837, .705, .82, .661, .457, .765, .322,
               .248, .304, -0.291, -0.314, -0.377, .397, .294, .428, -0.075, .192, .224,
               .037, .155, -.104, .077, -.488, .009), ncol=3)
> # using targetT
> SPA <- matrix(c(rep(NA, 6), .7, .0, .7, rep(0,3), rep(NA, 7),
                 0, 0, NA, 0, rep(NA, 4)), ncol=3)
> xt <- targetT(A, Target=SPA)
> # using pst
> SPApst <- matrix(c(rep(0, 6), .7, .0, .7, rep(0,3), rep(0, 7),
                    0, 0, 0, 0, rep(0, 4)), ncol=3)
> SPAW <- matrix(c(rep(0, 6), rep(1, 6), rep(0, 7), 1, 1, 0, 1,
                    rep(0, 4)), ncol=3)
> xpst <- pstT(A, Target = SPApst, W = SPAW)
> max(abs(loadings(xt) - loadings(xpst)))
```

Note that convergence tables are identical for both methods.

Additional examples are available in the help pages of `GPFoblq` and `rotations`.

## References

- Bernaards, C. A., & Jennrich, R. I. (2005). Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis. *Educational and Psychological Measurement*, 65(5), 676–696. <https://doi.org/10.1177/0013164404272507>
- Browne, M.W. (1972). Orthogonal rotation to a partially specified target. *British Journal of Mathematical and Statistical Psychology*, 25(1), 115–120. <https://doi.org/10.1111/j.2044-8317.1972.tb00482.x>
- Fischer, R., & Fontaine, J. (2010). Methods for investigating structural equivalence. In D. Matsumoto, & F. van de Vijver (Eds.), *Cross-Cultural Research Methods in Psychology* (179–215). Cambridge University press. <https://doi.org/10.1017/CBO9780511779381.010>
- Nguyen, H. V., & Waller, N. G. (2022). Local minima and factor rotations in exploratory factor analysis. *Psychological Methods*. Advance online publication. <https://doi.org/10.1037/met0000467>