Package 'FMradio'

October 12, 2022

Type Package

Title Factor Modeling for Radiomics Data

Version 1.1.1

Maintainer Carel F.W. Peeters <cf.peeters@vumc.nl>

Author Carel F.W. Peeters [cre, aut], Caroline Ubelhor [ctb], Kevin Kunzmann [ctb]

Description

Functions that support stable prediction and classification with radiomics data through factor-analytic modeling. For details, see Peeters et al. (2019) <arXiv:1903.11696>.

Depends R (>= 2.15.1)

biocViews

Imports stats, ggplot2, reshape, Biobase, graphics, expm, MASS

ByteCompile true

KeepSource yes

License GPL (>= 2)

BuildManual yes

URL https://github.com/CFWP/FMradio

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-16 12:10:04 UTC

R topics documented:

`FMradio-package` *Factor modeling for radiomic data*

Description

The `FMradio` package provides a workflow that uses factor modeling to project the high-dimensional and collinear radiomic feature-space onto a lower-dimensional orthogonal meta-feature space that retains most of the information contained in the full data set. These projected meta-features can be directly used as robust and stable covariates in any downstream prediction or classification model.

Details

Radiomics refers to the mining of large numbers of quantitative features from standard-of-care clinical images. `FMradio` aims to provide support for stable prediction and classification modeling with radiomics data, irrespective of imaging modality (such as MRI, PET, or CT). The workflow has 3 main steps that ultimately enable stable prediction and classification.

Step 1: Regularized correlation matrix estimation. Radiomic data are often high-dimensional in the sense that there are more features than observations. Moroever, radiomic data are often highly collinear, in the sense that collections of features may be highly correlated (in the absolute sense). This results in the correlation matrix on the radiomic features to be ill-conditioned or even singular. It is also this combination of characteristics that proves difficult to predictive modeling. As the factor-analytic procedure is based on the modeling of moment structures such as the correlation matrix, the first step is to obtain a regularized, well-conditioned estimate of the correlation matrix. The following functions are then of use:

- `radioHeat`
- `RF`
- `subSet`
- `regcor`

The `radioHeat` function can be used to visualize (a possibly regularized) correlation matrix as a heatmap. It can also be used to visually assess feature-redundancy. The `RF` function provides functionality for filtering features that are so collinear that they are deemed redundant. The `suBSet` function provides functionality to subset data objects to those features retained after possible filtering. The `regcor` function subsequently provides a regularized estimate of the correlation matrix (on the possibly filtered feature set).

Step 2: Factor analytic data compression. The next step would be to project the collinear and high-dimensional radiomic feature-space onto a lower-dimensional orthogonal meta-feature space. Factor analysis can be used for this purpose. The following functions are then of use:

- SA
- dimGB
- dimVAR
- SMC
- mlFA

The SA function assesses if performing a factor analysis on the (possibly regularized) correlation matrix would be appropriate. The dimGB function can be used to determine the number of latent factors (i.e., to determine the intrinsic dimensionality of the meta-feature space). The dimVAR and dimSMC functions can be used to provide additional decision support with respect to the output of the dimGB function. The mlFA function then performs a maximum likelihood factor analysis using the (possibly regularized) correlation matrix and the choice of intrinsic dimensionality as inputs.

Step 3: Obtaining factor scores. The third step would be to use the factor analytic solution to obtain factor scores: the score each object/individual would obtain on each of the latent factors. The following functions are then of use:

- facScore
- facSMC

The facScore function provides several options for computing factors scores. The determinacy of these scores can be assessed with the facSMC function.

Step 4: Prediction and classification. The factor scores obtained with Step 3 can be directly used as (low-dimensional and orthogonal) covariates in any prediction, classification or learning procedure. One may use the full flexibility provided by the CRAN repository for this step.

Additional functionality. The package also provides additional functionality. These are contained in the following (convenience) functions:

- dimLRT
- dimIC
- FAsim

The dimLRT and dimIC functions provide alternative options for assessing the number of latent factors using likelihood ratio testing and information criteria, respectively. These are only recommended when the sample size is large relative to the number of features. FAsim provides a flexible function for generating data according to the orthogonal common factor analytic model. All these functions may be of use in comparative exercises. The package also provides a wrapper function that automates the 3 main steps of the workflow:

- autoFMradio

Author(s)

Carel F.W. Peeters [cre, aut]
Caroline Ubelhor [ctb]
Kevin Kunzmann [ctb]

Maintainer: Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

`autoFMradio` *Wrapper for automated workflow*

Description

`autoFMradio` is a wrapper function that automates the three main steps of the `FMradio` workflow.

Usage

```
autoFMradio(X, t = .95, fold = 5, GB = 1, type = "thomson",
            verbose = TRUE, printInfo = TRUE, seed = NULL)
```

Arguments

`X`	A data `matrix` or an `ExpressionSet` object.
`t`	A scalar `numeric` indicating the absolute value for thresholding.
`fold`	A `numeric` integer or `integer` indicating the number of folds to use in cross-validation.
`GB`	A `numeric` integer or `integer` indicating which Guttman bound to use for determining the number of latent features to retain. Must be either 1, 2, or 3.
`type`	A `character` indicating the type of factor score to calculate. Must be one of: "thomson", "bartlett", "anderson".
`verbose`	A `logical` indicating if function should run silently. Runs silently when `verbose = FALSE`.
`printInfo`	A `logical` indicating if additional information should be printed on-screen. Suppresses printing when `verbose = FALSE`.
`seed`	A `numeric` integer or `integer` indicating the seed for the random number generator.

Details

The autoFMradio function automates the three main steps of the workflow by providing a wrapper around all core functions.

Step 1 (regularized correlation matrix estimation) is performed using the X, t, and fold arguments. The raw correlation matrix based on data X is redundancy-filtered using the threshold provided in t. Subsequently, a regularized estimate of the correlation matrix (on the possibly filtered feature set) is computed with the optimal penalty value determined by cross-validation. The number of folds is set by the fold argument. For more information on Step 1 see RF, subSet, and regcor.

Step 2 (factor analytic data compression) is performed using the GB argument. With this argument one can use either the first, second, or third Guttman bound to select the intrinsic dimensionality of the latent vector. This bound, together with the regularized correlation matrix, is used in a maximum likelihood factor analysis with simple-structure rotation. For more information on Step 2, see dimGB and mlFA.

Step 3 (obtaining factor scores) is performed using the type argument. It determines factor scores: the score each object/individual would obtain on each of the latent factors. The type argument determines the type of factor score that is calculated. For more information on Step 3, see facScore.

When printInfo = TRUE additional information is printed on-screen after the full procedure has run its course. This additional information pertains to each of the steps mentioned above. For Step 1 it reiterates the thresholding value for redundancy filtering and gives the number of features retained after this filtering. It also reiterates the number of folds used in determining the optimal penalty value as well as this value itself. Moreover, it provides the value of the Kaiser-Meyer-Olkin index on the optimal regularized correlation matrix estimate (see SA). For Step 2 it reiterates which Guttman bound was used in determining the number of latent factors as well as the number of latent factors retained. It also gives the proportion of explained variance under the factor solution of the chosen latent dimension (see dimVAR). For step 3 it reiterates the type of factor score that was calculated. Also, it prints the lowest 'determinacy score' amongst the latent factors (see facSMC).

The factor scores in the $Scores slot of the output (see below) can be directly used as input features in any prediction or classification procedure. In case of external (rather than internal) validation one can use the parameter matrices in the $Loadings and $Uniqueness slots in combination with fresh data to provide a validation factor projection based on the training solution. See Peeters *et al.* (2019).

Value

The function returns an object of class list:

$Scores	An object of class data.frame containing the factor scores. Observations are represented in the rows. Each column represent a latent factor.
$FilteredData	Subsetted data matrix containing only those features retained after redundancy filtering.
$FilteredCor	A correlation matrix based on the data in the $FilteredData slot.
$optPen	A numeric scalar representing the optimal value for the penalty parameter.
$optCor	A matrix representing the regularized correlation matrix under the optimal penalty-value.
$m	An integer correspond to number of latent factors retained under the chosen Guttman bound.

`$Loadings`	A matrix of class `loadings` representing the loadings matrix in which in which each element λ_{jk} is the loading of the jth feature on the kth latent factor.
`$Uniqueness`	A `matrix` representing the diagonal matrix carrying the unique variances.
`$Exvariance`	A `numeric` vector representing the cumulative variance for each respective latent feature.
`$determinacy`	A `numeric` vector indicating, for each factor, the squared multiple correlation between the observed features and the common latent factor.
`$used.seed`	A `numeric` or `integer` used as the starting seed in random number generation.

Note

When `seed = NULL` the starting seed is determined by drawing a single integer from the integers `1:9e5`. This non-user-supplied seed is also found in the `$used.seed` slot of the output.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

`RF`, `subSet`, `regcor`, `dimGB`, `mlFA`, `facScore`

Examples

```
## Simulate some data according to a factor model with 3 latent factors
simDAT <- FAsim(p = 24, m = 3, n = 40, loadingvalue = .9)
X <- simDAT$data

## Perform the lot
FullMonty <- autoFMradio(X, GB = 1, seed = 303)
```

---

dimGB *Assess the latent dimensionality using Guttman bounds*

Description

`dimGB` is a function that calculates the first, second, and third Guttman (lower-)bounds to the dimensionality of the latent vector. These can be used to choose the number of latent factors.

Usage

```
dimGB(R, graph = TRUE, verbose = TRUE)
```

Arguments

R	(Regularized) correlation matrix.
graph	A logical indicating if the results should be visualized.
verbose	A logical indicating if the function should run silently. Runs silently when verbose = FALSE.

Details

The communality in factor analysis refers to the amount of variance (of feature j) explained by the latent features. The correlation of any feature with itself can then be decomposed into common variance (the communality) and unique variance. This implies that unity (1) minus the unique variance for feature j equals the communality for feature j. From the matrix perspective one can then construct a reduced correlation matrix: the correlation matrix with communalities in the diagonal. This reduced correlation matrix is, by the assumptions on the factor model, Gramian and of rank m, with m indicating the intrinsic dimensionality of the latent vector. The dimension of the latent vector (i.e., the number of common factors) can then be assessed by evaluating the rank of the sample correlation matrix in which the diagonal elements are replaced with appropriate communality estimates.

In our case, which is often high-dimensional, we use the regularized correlation matrix as our sample-representation of the population correlation matrix. The diagonal elements are then replaced with Guttman's lower-bound estimates for the communalities (Guttman, 1956). Guttman (1956) gives 3 (ordered) lower-bound estimates. The first estimate is the most conservative, using 0 as a lower-bound estimate of the communalities. From this perspective, every positive eigenvalue of the reduced sample correlation matrix is indicative of a latent factor whose contribution to variance-explanation is above and beyond mere unique variance. The decisonal approach would then be to retain all such factors. See Peeters *et al.* (2019) for additional detail.

The Guttman approach has historically been used as a lower-bound estimate of the latent dimensionality. We consider the decisional approach stated above to give an upper-bound. Peeters *et al.* (2019) contains an extensive simulation study showing that in high-dimensional situations this decisional approach provides a reliable upper-bound. The choice of the number of factors can be further assessed with the SMC and dimVAR functions. Assessments provided by these latter functions may inform if the result of the decisional rule above should be accepted or be treated as an upper-bound.

When graph = TRUE the Guttman bounds are visualized. It plots the consecutive eigenvalues for each of the reduced correlation matrices. The number of positive eigenvalues for each respective reduced correlation matrix then corresponds to each of the respective Guttman bounds. The visualization may be of limited value when the feature-dimension gets (very) large.

Value

The function returns an object of class table. The entries correspond to the first, second, and third Guttman bounds.

Note

- Again, from a historical perspective, the decisional rule would have been used as a lower-bound to the question of the number of latent common factors. In high-dimensional situations we recommend to use it as an upper-bound.

- Other functions for factor analytic dimensionality assessment are dimIC and dimLRT. In high-dimensional situations usage of dimGB is recommended over these other functions.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Guttman, L. (1956). Best possible systematic estimates of communalities. Psychometrika, 21:273–285.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

SMC, dimVAR, FAsim

Examples

```
## Simulate some data according to a factor model with 5 latent factors
## $cormatrix gives the correlation matrix on the generated data
simDAT <- FAsim(p = 50, m = 5, n = 100)
simDAT$cormatrix

## Evaluate the Guttman bounds
## First Guttman bound indicates to retain 5 latent factors
GB <- dimGB(simDAT$cormatrix)
print(GB)
```

---

dimIC *Assess the latent dimensionality using information criteria*

Description

A function that calculates either the AIC or the BIC on the factor model. These can be used to choose the number of latent factors.

Usage

```
dimIC(R, n, maxdim, Type = "BIC", graph = TRUE, verbose = TRUE)
```

Arguments

R	(Regularized) correlation matrix.
n	A numeric scalar representing the sample size.
maxdim	A numeric integer or integer indicating the maximum factor dimension to be assessed.

Type	A character indicating the type of IC to be calculated. Must be one of: "AIC", "BIC".
graph	A logical indicating if the results should be visualized.
verbose	A logical indicating if the function should run silently. Runs silently when verbose = FALSE.

Details

Information criteria (IC) are often used in selecting the number of latent factor to retain. IC aim to balance model fit with model complexity. They evaluate (minus 2 times) the maximized value of the (model-dependent) likelihood function weighed with a penalty function that is dependent on the free parameters in the model. Different penalizations define the different types of IC. The strategy would be to determine IC scores for a range of consecutive values of the latent factor dimension. This function then determines scores for factor solutions ranging from 1 to maxdim latent factors. The solution with the lowest IC score is deemed optimal. The function allows for the calculation of either the Akaike information criterion (AIC; Akaike, 1973) or the Bayesian information criterion (BIC; Schwarz, 1978). Also see the Supplementary Material of Peeters *et al.* (2019) for additional detail.

When graph = TRUE the IC scores are visualized. The graph plots the IC score against the consecutive dimensions of the factor solution.

Value

The function returns an object of class data.frame. The first column represents the assessed dimensions running from 1 to maxdim. The second column represents the corresponding values of the chosen information criterion.

Note

- The argument maxdim cannot exceed the Ledermann-bound (Ledermann, 1937): $\lfloor [2p + 1 - (8p + 1)^{1/2}]/2 \rfloor$, where p indicates the observed-feature dimension. Usually, one wants to set maxdim much lower than this bound.
- Other functions for factor analytic dimensionality assessment are dimGB and dimLRT. In high-dimensional situations usage of dimGB on the regularized correlation matrix is recommended.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In: B. N. Petrov and F. Csaki (Eds.) Second International Symposium on Information Theory, pages 267–281. Budapest: Akademiai Kaido.

Ledermann, W. (1937). On the rank of the reduced correlational matrix in multiple factor analysis. Psychometrika, 2:85–93.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

Schwarz, G.E. (1978). Estimating the dimension of a model. Annals of Statistics, 6:461–464.

See Also

dimGB, FAsim

Examples

```
## Simulate some data according to the factor model
## $cormatrix gives the correlation matrix on the generated data
simDAT <- FAsim(p = 50, m = 5, n = 100)
simDAT$cormatrix

## Calculate the AIC for models of factor dimension 1 to 20
AIC <- dimIC(simDAT$cormatrix, n = 100, Type = "AIC", maxdim = 20)
print(AIC)

## Calculate the BIC for models of factor dimension 1 to 20
BIC <- dimIC(simDAT$cormatrix, n = 100, Type = "BIC", maxdim = 20)
print(BIC)
```

---

dimLRT *Assess the latent dimensionality using a likelihood ratio test*

Description

dimLRT is a function that evaluates a likelihood ratio test on the factor model. It can be used to choose the number of latent factors.

Usage

```
dimLRT(R, X, maxdim, rankDOF = TRUE, graph = TRUE,
       alpha = .05, Bartlett = FALSE, verbose = TRUE)
```

Arguments

R	(Regularized) correlation matrix.
X	A (possibly centered and scaled and possibly subsetted) data matrix.
maxdim	A numeric integer or integer indicating the maximum factor dimension to be assessed.
rankDOF	A logical indicating if the degrees of freedom should be based on the rank of the raw correlation matrix.
graph	A logical indicating if the results should be visualized.
alpha	A numeric scalar representing the alpha level. Only used when graph = TRUE.
Bartlett	A logical indicating if the Bartlett correction should be applied.
verbose	A logical indicating if the function should run silently. Runs silently when verbose = FALSE.

Details

The most formal approach to factor analytic dimensionality assessment is through likelihood ratio (LR) testing. The basic idea is to test the m-factor model against the saturated model. The corresponding LR criterion then converges, under the standard correlation matrix and corresponding parameter estimates under m-factors, to $(n-1)$ times a certain discrepancy function evaluated at the maximum-likelihood-parameters under the m-factor model. This quantity is approximately χ^2-distributed under certain regularity conditions (Amemiya & Anderson, 1990). The general strategy would then be to sequentially test solutions of increasing dimensionality $m = 1, \ldots,$ maxdim until the null hypothesis (stating that the m-factor model holds) is *not* rejected at Type-I error level `alpha`.

The degrees of freedom for the LRT under the m-factor model equals the number of parameters in the saturated model (i.e., the unstructured sample correlation) minus the number of freely estimable parameters in the m-factor model. Note that the general stategy above makes use of asymptotic results. In our setting, however, the observation dimension (n) is usually small relative to the feature dimension (p). Hence, the standard test will in a sense overestimate the degrees of freedom. One simple option dealing with this observation would be to adapt the degrees of freedom to incorporate the rank deficiency of `R`. This road is taken when `rankDOF = TRUE`. Bartlett (1950) proposed a correction factor when the sample size is small to make the test statistic behave more χ^2-like. This correction factor is used when `Bartlett = TRUE`.

When `graph = TRUE` the LRT results are visualized. The graph plots the LRT p-values against the consecutive dimensions of the factor solution. A horizontal line is plotted at the value provided in the `alpha` argument.

Unless the number of observations is much larger than the number of features, the LRT is not recommended for inference in general. In Peeters *et al.* (2019) the LRT was assessed in a comparative setting inviolving high-dimensional factor models.

Value

The function returns an object of class `data.frame`. The first column represents the assessed dimensions running from 1 to `maxdim`. The second column represents the observed values of the LRT statistic. The third column represents the corresponding p-values.

Note

- Note that, for argument `X`, the observations are expected to be in the rows and the features are expected to be in the columns.
- The argument `maxdim` cannot exceed the Ledermann-bound (Ledermann, 1937): $\lfloor [2p + 1 - (8p+1)^{1/2}]/2 \rfloor$, where p indicates the observed-feature dimension. Usually, one wants to set `maxdim` much lower than this bound.
- note that, if $p > n$, then the maximum rank of the raw correlation matrix is $n - 1$. In this case there is an alternative Ledermann-bound when `rankDOF = TRUE`. The number of information points in the correlation matrix is then given as $n \times (n-1)/2$ and this number must exceed $p \times \text{maxdim} + p - (\text{maxdim} \times (\text{maxdim} - 1))/2$, putting more restrictions on `maxdim`.
- Other functions for factor analytic dimensionality assessment are `dimGB` and `dimIC`. In high-dimensional situations usage of `dimGB` on the regularized correlation matrix is recommended.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>, Caroline Ubelhor

References

Amemiya, Y., & Anderson, T.W. (1990). Asymptotic chi-square tests for a large class of factor analysis models. The Annals of Statistics, 18:1453–1463.

Bartlett, M.S. (1950). Tests of significance in factor analysis. British Journal of Psychology (Statistics Section), 3:77–85.

Ledermann, W. (1937). On the rank of the reduced correlational matrix in multiple factor analysis. Psychometrika, 2:85–93.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

dimGB, FAsim

Examples

```
## Simulate some data according to the factor model
## $cormatrix gives the correlation matrix on the generated data
simDAT <- FAsim(p = 50, m = 5, n = 500)
simDAT$cormatrix

## Calculate the LRT for models of factor dimension 1 to 20
LRT <- dimLRT(simDAT$cormatrix, simDAT$data, maxdim = 20, rankDOF = FALSE)
print(LRT)
```

---

dimVAR *Assessing variances under factor solutions*

Description

dimVAR is a support function that assesses the proportion of and cumulative variances for a range of factor solutions.

Usage

```
dimVAR(R, maxdim, graph = TRUE, verbose = TRUE)
```

Arguments

R	(Regularized) correlation matrix.
maxdim	A numeric integer or integer indicating the maximum factor dimension to be assessed.
graph	A logical indicating if the results should be visualized.
verbose	A logical indicating if the function should run silently. Runs silently when verbose = FALSE.

Details

To assess a factor solution under m factors one might look at the proportion of explained variance. The `dimVAR` function calculates the proportion of variance explained by any factor as well as the proportion of variance explained by all factors for each factor solution raging from 1 to `maxdim`. Qualitatively, we want the proportion of variance explained by all factors to be appreciable (rules of thumb would say in excess of 70%). Moreover, one would want the proportion of variance explained by the kth factor in relation to the $(k-1)$th factor to be appreciable and the proportion of variance of the $(k+1)$th factor in relation to the kth factor to be negligible.

When `graph = TRUE` also a graph is returned visualizing the total cumulative variance against the dimension of the factor solution. Hence, it plots the total cumulative variances against the respective factor solutions ranging from 1 to `maxdim`. The point at which the graph flattens out is indicative of a formative number of latent factors.

Value

Returns an object of class `list`.

`$CumVar` Contains a `numeric` vector with the cumulative variances explained for each factor solution from 1 to `maxdim`.

`$varianceTables`
This slot is itself a `list`. It contains, for each factor solution, a matrix with the sum of squares (SS), proportion variance (PV), and cumulative variance (CV) for each respective latent feature. Say one wants to access the variance table for a solution under 5-factors. Then one can call `` $varianceTables$`dimension = 5` ``. Similar calls are made to retrieve the variance table for other factor solutions.

Note

- The argument `maxdim` cannot exceed the Ledermann-bound (Ledermann, 1937): $\lfloor [2p + 1 - (8p + 1)^{1/2}]/2 \rfloor$, where p indicates the observed-feature dimension. Usually, one wants to set `maxdim` much lower than this bound.
- The tabulations in the `$varianceTables` slot are based on unrotated maxmimum likelihood factor solutions. Note that the total cumulative variance does not depend on the choice of (orthogonal) rotation.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Ledermann, W. (1937). On the rank of the reduced correlational matrix in multiple factor analysis. Psychometrika, 2:85–93.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

`dimGB`, `FAsim`, `mlFA`, `SMC`

Examples

```
## Simulate some high-dimensional data according to the factor model
simDAT <- FAsim(p = 50, m = 5, n = 40)

## Regularize the correlation matrix
RegR <- regcor(simDAT$data)

## Assess proportion and cumulative variances for a range of factor solutions
## Inspect, for example, the variance table for the 5-factor solution
V <- dimVAR(RegR$optCor, maxdim = 20)
V$varianceTables$`dimension = 5`
```

---

facScore *Compute factor scores*

Description

facScore is a function that computes factor scores, the score each person/object attains on each latent factor.

Usage

```
facScore(X, LM, UM, type = "thomson")
```

Arguments

X	A (scaled and possibly subsetted) data matrix.
LM	A (rotated) loadings matrix. Usually the $Loadings-slot object from the mlFA function output.
UM	A diagonal uniquenesses matrix. Usually the $Uniqueness-slot object from the mlFA function output.
type	A character indicating the type of factor score to calculate. Must be one of: "thomson", "bartlett", "anderson".

Details

Once a factor model is fitted one may desire an estimate of the score each object/individual would obtain on each of the latent factors. Such scores are referred to as factor scores. The facScore function provides several types of factor score estimates. The default are Thomson-type scores (Thomson, 1939). These may be viewed as (empirical) Bayesian-type scores. Bartlett-type scores (Bartlett, 1937) are unbiased but less efficient in terms of mean-squared error. Under the orthogonal model the latent factors are orthogonal in the population and, hence, the Thomson and Bartlett-type factor scores will be near orthogonal in the sample. Anderson and Rubin (1956) constructed an alternative estimator for the factor scores that enforces their orthogonality in the sample.

Value

The function returns an object of class `data.frame`. Observations are represented in the rows. Each column represent a latent factor.

Note

The input data (argument `X`) are assumed to be scaled (or at least centered). The `UM` matrix is assumed to be positive definite. The `LM` matrix is assumed to be of full column rank.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Anderson, T.W., & Rubin, H. (1956). Statistical inference in factor analysis. In Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, volume 5: Contributions to Econometrics, Industrial Research, and Psychometry, pages 111–150. Berkeley, CA: University of California Press.

Bartlett, M.S. (1937). The statistical conception of mental factors. British Journal of Psychology, 28:97–104.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

Thomson, G. (1939). The Factorial Analysis of Human Ability. London: University of Londen Press.

See Also

`dimGB`, `mlFA`, `facSMC`

Examples

```
## Simulate some data according to a factor model with 5 latent factors
## Simulate high-dimensional situation in the sense that p > n
## $cormatrix gives the correlation matrix on the generated data
simDAT <- FAsim(p = 50, m = 5, n = 40, loadingvalue = .9)
simDAT$cormatrix

## Regularize the correlation matrix
RegR <- regcor(simDAT$data)

## Evaluate the Guttman bounds
## First Guttman bound indicates to retain 5 latent factors
GB <- dimGB(RegR$optCor)
print(GB)

## Produce ML factor solution under 5 factors
## Print loadings structure of this solution
fit <- mlFA(RegR$optCor, 5)
print(fit$Loadings, digits = 2, cutoff = .3, sort = TRUE)
```

```
## Obtain factor-scores
scores <- facScore(scale(simDAT$data), fit$Loadings, fit$Uniqueness)
print(scores)
```

`facSMC` *Evaluate the determinacy of factor scores*

Description

`facSMC` is a function with which one may evaluate the determinacy of factor scores.

Usage

```
facSMC(R, LM)
```

Arguments

`R`	(Regularized) correlation `matrix`.
`LM`	A (rotated) loadings `matrix`. Usually the `$Loadings`-slot object from the `mlFA` function output.

Details

The `facSMC` function calculates the squared multiple correlations between the observed features and the common latent factors. The closer to unity, the lesser the problem of factor-score indeterminacy and the better one is able to uniquely determine the factor scores. In practice, a squared multiple correlation equalling or exceeding .9 would be considered adequate. See Mulaik (2010, Chapter 13) and Peeters *et al.* (2019, Supplementary Materials) for further details.

Value

The function returns a `numeric` vector indicating, for each factor, the squared multiple correlation between the observed features and the common latent factor.

Note

Note that the computations assume an orthogonal factor model. Hence, only orthogonal rotations of the loadings matrix should be used (or no rotation at all).

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Mulaik, S.A. (2010). Foundations of Factor Analysis. Boca Raton: Chapman & Hall/CRC, 2nd edition.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

facScore

Examples

```
## Simulate some data according to a factor model with 5 latent factors
## Simulate high-dimensional situation in the sense that p > n
## $cormatrix gives the correlation matrix on the generated data
simDAT <- FAsim(p = 50, m = 5, n = 40, loadingvalue = .9)
simDAT$cormatrix

## Regularize the correlation matrix
RegR <- regcor(simDAT$data)

## Evaluate the Guttman bounds
## First Guttman bound indicates to retain 5 latent factors
GB <- dimGB(RegR$optCor)
print(GB)

## Produce ML factor solution under 5 factors
## Print loadings structure of this solution
fit <- mlFA(RegR$optCor, 5)
print(fit$Loadings, digits = 2, cutoff = .3, sort = TRUE)

## Obtain factor-scores
scores <- facScore(scale(simDAT$data), fit$Loadings, fit$Uniqueness)
print(scores)

## Evaluate determinacy of factor scores
fd <- facSMC(RegR$optCor, fit$Loadings)
print(fd)
```

FAsim *Simulate data according to the common factor analytic model*

Description

FAsim is a function that enables the simulation of data according to the common factor analytic model.

Usage

```
FAsim(p, m, n, simplestructure = TRUE, balanced = TRUE,
      loadingfix = TRUE, loadingnegative = TRUE,
      loadingvalue = .8, loadingvaluelow = .2, numloadings,
      loadinglowerH = .7, loadingupperH = .9,
      loadinglowerL = .1, loadingupperL = .3)
```

Arguments

p	A numeric integer or integer indicating the number of observed features.
m	A numeric integer or integer indicating the latent dimension of the factor solution (i.e., the number of factors).
n	A numeric integer or integer indicating the number of samples.
simplestructure	A logical indicating if the generating factor structure should be factorially pure.
balanced	A logical indicating if the high (i.e., qualitatively 'significant') loadings should be divided evenly over the respective factors.
loadingfix	A logical indicating if the loadings should have a fixed value.
loadingnegative	A logical indicating if, next to positive, also negative loadings should be present.
loadingvalue	A numeric indicating the value for high (i.e., qualitatively 'significant') loadings. Used when loadingfix = TRUE.
loadingvaluelow	A numeric indicating the value for low loadings. Used when loadingfix = TRUE & simplestructure = FALSE.
numloadings	A numeric vector with length equalling argument m, indicating the number of high (i.e., qualitatively 'significant') loadings per factor. Used when balanced = FALSE.
loadinglowerH	A numeric indicating the lower-bound of high (i.e., qualitatively 'significant') loadings. Used when loadingfix = FALSE.
loadingupperH	A numeric indicating the upper-bound of high (i.e., qualitatively 'significant') loadings. Used when loadingfix = FALSE.
loadinglowerL	A numeric indicating the lower-bound of low (i.e., qualitatively 'non-significant') loadings. Used when loadingfix = FALSE & simplestructure = FALSE.
loadingupperL	A numeric indicating the upper-bound of low (i.e., qualitatively 'non-significant') loadings. Used when loadingfix = FALSE & simplestructure = FALSE.

Details

FAsim provides certain flexibility when generating data according to an orthogonal common factor-analytic model. It can produce data according to, for example, (i) factorially pure loadings structures, (ii) loadings-structures with only positive entries or both positive and negative loadings, (iii) loadings-structures with fixed values or varying values, (iv) balanced and unbalanced loadings-structures.

Value

The function returns an object of class list:

$data	A standardized data matrix of size $n \times p$.
$loadings	Loadings matrix of size $p \times m$ on which the data-generation was based.

`$Uniqueness`	A `numeric` vector of size p representing the uniquenesses on which the data-generation was based.
`$cormatrix`	A $p \times p$ correlation `matrix` based on the generated data in slot `$data`.

Note

- A uniform distribution is assumed when generating draws between `loadinglowerH` and `loadingupperH`.
- A uniform distribution is assumed when generating draws between `loadinglowerL` and `loadingupperL`.
- The argument `m` cannot exceed the Ledermann-bound (Ledermann, 1937): $\lfloor [2p + 1 - (8p + 1)^{1/2}]/2 \rfloor$, where p indicates the observed-feature dimension.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Ledermann, W. (1937). On the rank of the reduced correlational matrix in multiple factor analysis. Psychometrika, 2:85–93.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

`dimGB`, `mlFA`, `facScore`

Examples

```
## Simulate some data according to a factor model with 3 latent factors
## Balanced and factorially pure loadings structure
simDAT <- FAsim(p = 24, m = 3, n = 40, loadingvalue = .9)
simDAT$loadings

## Simulate some data according to a factor model with 3 latent factors
## Unbalanced and factorially pure loadings structure
simDAT <- FAsim(p = 24, m = 3, n = 40, loadingvalue = .9,
                balanced = FALSE, numloadings = c(10,10,4))
simDAT$loadings

## Simulate some data according to a factor model with 3 latent factors
## Unbalanced and factorially non-pure loadings structure
simDAT <- FAsim(p = 24, m = 3, n = 40, loadingvalue = .9,
                balanced = FALSE, numloadings = c(10,10,4),
                simplestructure = FALSE)
simDAT$loadings

## Simulate some data according to a factor model with 3 latent factors
## Unbalanced and factorially non-pure loadings structure
## Non-fixed high and low loadings
simDAT <- FAsim(p = 24, m = 3, n = 40, loadingvalue = .9,
                balanced = FALSE, numloadings = c(10,10,4),
```

```
                simplestructure = FALSE, loadingfix = FALSE)
simDAT$loadings
```

`mlFA` *Maximum likelihood factor analysis*

Description

`mlFA` is a function that performs a maximum likelihood factor analysis.

Usage

```
mlFA(R, m)
```

Arguments

`R`	(Regularized) correlation `matrix`.
`m`	A `numeric` integer or `integer` indicating the latent dimension of the factor solution (i.e., the number of factors).

Details

This function is basically a wrapper around the factanal function from the stats package. Its purpose is to produce a factor solution of the chosen dimension (argument `m`) by a maximum likelihood estimation procedure (Joreskog, 1967). The wrapper ensures that the model is fitted under the same circumstances under which latent dimensionality is assessed with functions such as `dimLRT` and `dimIC`. The function produces a Varimax rotated (Kaiser, 1958) factor solution. The output can be used to produce factor scores by the `facScore` function.

Value

The function returns an object of class `list`:

`$Loadings`	A matrix of class `loadings` representing the loadings matrix in which in which each element λ_{jk} is the loading of the jth feature on the kth latent factor.
`$Uniqueness`	A `matrix` representing the diagonal matrix carrying the unique variances.
`$rotmatrix`	A `matrix` representing the Varimax rotation matrix.

Note

- Note that the order of the features in the `$Loadings` and `$Uniqueness` slots of the output is determined by the order of the features for the input argument `R`. As the `$Loadings` slot gives an object of class "loadings" it can be subjected to the `print` function, which sorts the output to emphasize the loadings structure when calling `sort = TRUE`.
- Note that the maximum likelihood procedure is stable when a regularized correlation matrix is used as the input for argument `R`.
- In high-dimensional situations usage of `dimGB` on the regularized correlation matrix is recommended to determine the value for argument `m`.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Joreskog, K.G (1967). Some contributions to maximum likelihood factor analysis. Psychometrika, 32:443–482.

Kaiser, H.F. (1958). The varimax criterion for analytic rotation in factor analysis. Psychometrika, 23:187–200.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

dimGB, facScore

Examples

```
## Simulate some data according to a factor model with 5 latent factors
## Simulate high-dimensional situation in the sense that p > n
## $cormatrix gives the correlation matrix on the generated data
simDAT <- FAsim(p = 50, m = 5, n = 40, loadingvalue = .9)
simDAT$cormatrix

## Regularize the correlation matrix
RegR <- regcor(simDAT$data)

## Evaluate the Guttman bounds
## First Guttman bound indicates to retain 5 latent factors
GB <- dimGB(RegR$optCor)
print(GB)

## Produce ML factor solution under 5 factors
## Print loadings structure of this solution
fit <- mlFA(RegR$optCor, 5)
print(fit$Loadings, digits = 2, cutoff = .3, sort = TRUE)
```

---

radioHeat *Visualize a (correlation) matrix as a heatmap*

Description

radioHeat is a function that provides dedicated heatmapping of a radiomics-based correlation matrix It can be used to visually assess the elements of a (possibly thresholded) matrix. It also supports the assessment of collinearity.

Usage

```
radioHeat(R, lowColor = "blue", highColor = "red", labelsize = 10,
          diag = TRUE, threshold = FALSE, threshvalue = .95,
          values = FALSE, textsize = 10, legend = TRUE, main = "")
```

Arguments

R	(regularized) correlation matrix
lowColor	A character that determines the color scale in the negative range.
highColor	A character that determines the color scale in the positive range.
labelsize	A numeric that sets the textsize of row and column labels.
diag	A logical determining if the diagonal elements of the matrix should be included in the color scaling. This argument is only used when R is a square matrix.
threshold	A logical determining if only values above a certain (absolute) threshold should be visualized.
threshvalue	A numeric indicating the absolute thresholding value when threshold = TRUE.
values	A logical determining the optional inclusion of cell-values.
textsize	A numeric indicating the textsize of the cell-values when values = TRUE.
legend	A logical indicating whether a color legend should be included.
main	A character giving the main figure title.

Details

This function utilizes ggplot2 (Wickham, 2009) to visualize a matrix as a heatmap: a false color plot in which the individual matrix entries are represented by colors. lowColor determines the color scale for matrix entries in the negative range. highColor determines the color scale for matrix entries in the positive range. For the colors supported by the arguments lowColor and highColor, see https://stat.columbia.edu/~tzheng/files/Rcolor.pdf. White entries in the plot represent the midscale value of 0. One can opt to set the diagonal entries to the midscale color of white when one is interested in (heatmapping) the off-diagonal elements only. To achieve this, set diag = FALSE. Naturally, the diag argument is only used when the input matrix M is a square matrix.

The intended usage is to visualize a correlation matrix on radiomic features as a heatmap. Such a heatmap may be used to support the assessment of strong collinearity or even redundancy amongst the features. To this end, it is also possible to visualize a thresholded correlation matrix when threshold = TRUE based on the absolute thresholding value given in the threshvalue argument (hence the thresholding is done internally). This enables easier visual access to (blocks of) collinearity in radiomic-feature-based correlation matrices.

Note

- While geared towards the visualization of correlation matrices, the function is quite general, in the sense that it can represent any matrix as a heatmap.
- When values = TRUE and threshold = TRUE the cell-values are those of the thresholded matrix.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Wickham, H. (2009). ggplot2: elegant graphics for data analysis. New York: Springer.

See Also

RF, regcor

Examples

```
## Generate some (high-dimensional) data
p = 25
n = 10
set.seed(333)
X = matrix(rnorm(n*p), nrow = n, ncol = p)
colnames(X)[1:25] = letters[1:25]
R <- cor(X)

## Visualize the correlation matrix as a heatmap
radioHeat(R)

## Remove diagonal entries from visualization
radioHeat(R, diag = FALSE)

## Additionally, visualize only those entries whose absolute value exceed .5
radioHeat(R, diag = FALSE, threshold = TRUE, threshvalue = .5)

## Additionally, include cell values
radioHeat(R, diag = FALSE, threshold = TRUE, threshvalue = .5,
          values = TRUE, textsize = 3)
```

---

regcor *Regularized correlation matrix estimation*

Description

regcor is a function that determines the optimal penalty value and, subsequently, the optimal Ledoit-Wolf type regularized correlation matrix using K-fold cross validation of the negative log-likelihood.

Usage

```
regcor(X, fold = 5, verbose = TRUE)
```

Arguments

X	A (possibly centered and scaled and possibly subsetted) data matrix.
fold	A numeric integer or integer indicating the number of folds to use in cross-validation.
verbose	A logical indicating if function should run silently. Runs silently when verbose = FALSE.

Details

This function estimates a Ledoit-Wolf-type (Ledoit & Wolf, 2004) regularized correlation matrix. The optimal penalty-value is determined internally by K-fold cross-validation of the of the negative log-likelihood function. The procedure is efficient as it makes use of the Brent root-finding procedure (Brent, 1971). The value at which the K-fold cross-validated negative log-likelihood score is minimized is deemed optimal. The function employs the Brent algorithm as implemented in the optim function. It outputs the optimal value for the penalty parameter and the regularized correlation matrix under this optimal penalty value. See Peeters *et al.* (2019) for further details.

The optimal penalty-value can be used to assess the conditioning of the estimated regularized correlation matrix using, for example, a condition number plot (Peeters, van de Wiel, van Wieringen, 2016). The regularized correlation matrix under the optimal penalty can serve as the input to functions that assess factorability (SA), evaluate optimal choices of the latent common factor dimensionality (e.g., dimGB), and perform maximum likelihood factor analysis (mlFA).

Value

The function returns an object of class list:

$optPen	A numeric scalar representing the optimal value for the penalty parameter.
$optCor	A matrix representing the regularized correlation matrix under the optimal penalty-value.

Note

Note that, for argument X, the observations are expected to be in the rows and the features are expected to be in the columns.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Brent, R.P. (1971). An Algorithm with Guaranteed Convergence for Finding a Zero of a Function. Computer Journal 14: 422–425.

Ledoit, O, & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. Journal of Multivariate Analysis, 88:365–411.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

Peeters, C.F.W., van de Wiel, M.A., & van Wieringen, W.N. (2016). The spectral condition number plot for regularization parameter determination, arXiv:1608.04123v1 [stat.CO].

See Also

RF, subSet, SA, dimGB, mlFA

Examples

```
## Generate some (high-dimensional) data
## Get correlation matrix
p = 25
n = 10
set.seed(333)
X = matrix(rnorm(n*p), nrow = n, ncol = p)
colnames(X)[1:25] = letters[1:25]
R <- cor(X)

## Redundancy visualization, at threshold value .9
radioHeat(R, diag = FALSE, threshold = TRUE, threshvalue = .9)

## Redundancy-filtering of correlation matrix
Rfilter <- RF(R, t = .9)
dim(Rfilter)

## Subsetting data
DataSubset <- subSet(X, Rfilter)
dim(DataSubset)

## Obtain regularized correlation matrix
RegR <- regcor(DataSubset, fold = 5, verbose = TRUE)
RegR$optPen  ## optimal penalty-value
```

RF *Redundancy filtering of a square (correlation) matrix*

Description

RF is a function that performs redundancy filtering (RF) of a square (correlation) matrix.

Usage

```
RF(R, t = .95)
```

Arguments

R	Square (correlation) matrix.
t	A scalar numeric indicating the absolute value for thresholding.

Details

Radiomic features can be very strongly correlated. The sample correlation matrix on extracted radiomic features will then often display strong collinearity. The collinearity may be so strong as to imply redundant information, in the sense that some entries will approach perfect (negative) correlation. Hence, one may wish to perform redundancy-filtering on the raw sample correlation matrix in such situations.

The `RF` function uses an Algorithm from Peeters *et al.* (2019) to remove the minimal number of redundant features under absolute marginal correlation threshold `t`. We recommend setting $\mathtt{t} \in [.9, .95]$. Details of the algorithm can be found in Peeters *et al.* (2019).

The function returns a redundancy-filtered correlation `matrix`. This return output may subsequently be used in the `subSet` function. This is a convenience function that subsets a dataset to the features retained after redundancy-filtering.

Value

Returns a redundancy-filtered `matrix`.

Note

- While geared towards the redundancy filtering of correlation matrices, the function is quite general, in the sense that it can be used to filter any square `matrix`.
- When the input matrix `R` is a correlation matrix, then argument `t` should satisfy $-1 < \mathtt{t} < 1$, for the return matrix to be sensical for further analysis.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

`subSet`, `regcor`

Examples

```
## Generate some (high-dimensional) data
## Get correlation matrix
p = 25
n = 10
set.seed(333)
X = matrix(rnorm(n*p), nrow = n, ncol = p)
colnames(X)[1:25] = letters[1:25]
R <- cor(X)

## Redundancy visualization, at threshold value .9
radioHeat(R, diag = FALSE, threshold = TRUE, threshvalue = .9)
```

```
## Redundancy-filtering of correlation matrix
Rfilter <- RF(R, t = .9)
dim(Rfilter)
```

SA *Calculate the KMO measure of feature-sampling adequacy*

Description

SA is a function that calculates the Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy.

Usage

```
SA(R)
```

Arguments

R	(Regularized) covariance or correlation matrix.

Details

The SA function calculates the Kaiser-Meyer-Olkin (KMO) measure of feature-sampling adequacy (Kaiser & Rice, 1974). It provides a practical option for the assessment of factorability. Factorability refers to the assessment of the ability to identify coherent common latent factors from a given correlation matrix. In common factor analysis the observed features are assumed to be independent *given* the common latent features. Under this crucial model assumption, the inverse of the population correlation matrix is diagonal. Hence, to assess factorability one could assess if the inverse of the sample correlation matrix is near-diagonal. The KMO index provides for such an assessment by "comparing the sizes of the off-diagonal entries of the regularized correlation matrix to the sizes of the off-diagonal entries of its scaled inverse" (Peeters *et al.*, 2019). It takes values in $[0, 1]$ and larger values are preferred. A KMO index between .9 and 1 would be considered to be indicative of great factorability. For rules of thumb regarding interpretation of KMO index value, see Kaiser (1970). The SA function calculates an overall KMO index as well as the KMO index per observed feature.

The intended usage of the SA function is to assess if performing a factor analysis on a given (regularized) correlation matrix can be considered appropriate. As such, it succeeds usage of the regcor function (for high-dimensional and/or strongly collinear settings) and precedes usage of the dimGB and mlFA functions.

Value

The function returns an object of class list:

$KMO	A numeric scalar representing the overall KMO index.
$KMOfeature	A numeric vector giving the KMO index per feature.

Note

The input `matrix R` should be nonsingular for the KMO to be computed. When `R` is singular one may regularize it using the `regcor` function.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Kaiser, H.F. (1970). A second-generation little jiffy. Psychometrika, 35:401–415.

Kaiser, H.F., & Rice., J. (1974). Little jiffy, mark IV. Educational and Pscyhological Measurement, 34:111–117.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

`regcor`, `dimGB`, `mlFA`

Examples

```
## Generate some (high-dimensional) data
p = 25
n = 10
set.seed(333)
X = matrix(rnorm(n*p), nrow = n, ncol = p)
colnames(X)[1:25] = letters[1:25]

## Obtain regularized correlation matrix
RegR <- regcor(X, fold = 5, verbose = TRUE)

## Assess factorability through the KMO index
factorable <- SA(RegR$optCor)
factorable$KMO
factorable$KMOfeature
```

`SMC` *Compare squared multiple correlations with model-based communalities*

Description

`SMC` is a function that compares the best lower-bound estimates to the communalities with the model-based communalities implied by a factor solution of dimension m.

Usage

```
SMC(R, LM)
```

Arguments

R	(Regularized) correlation matrix.
LM	(Rotated) factor loadings matrix.

Details

This function can be used to qualitatively assess the choice of dimensionality (as well as the fit) in the m-factor model. This is done using the concept of communalities. The communality refers to the amount of variance of feature j explained by the latent features. It is then of interest to compare lower-bound estimates of the (population) communalities to the extracted communalities under the m-factor model.

Guttman (1956) gave the best possible lower-bound estimates to the communalities, which can essentially be considered squared multiple correlations: the proportion of variance in feature j that is explained by the remaining $p - 1$ features. To assess a factor model, these might be compared to the retrieved estimated communalities under the m-factor model. When the chosen latent dimensionality is sufficient then one would expect that, for almost all features, the retrieved communality approximately equals or exceeds its corresponding lower-bound estimate. If this is not the case then one might have extracted too few factors.

Value

The function returns a matrix. The first column (labeled 'SMC') contains the lower-bound estimates to the communalities. The second column (labeled 'Communalities') contains the retrieved estimated communalities under the m-factor model.

Note

Note that the choice of orthogonal rotation does not affect the model-implied communality estimates.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Guttman, L. (1956). Best possible systematic estimates of communalities. Psychometrika, 21:273–285.

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

dimGB, FAsim, mlFA, dimVAR

Examples

```
## Simulate some high-dimensional data according to the factor model
simDAT <- FAsim(p = 50, m = 5, n = 40)

## Regularize the correlation matrix
RegR <- regcor(simDAT$data)

## Fit 5-factor model to the regularized correlation matrix
fit <- mlFA(RegR$optCor, m = 5)

## Compare lower-bound estimates to communalities with model-implied ones
C <- SMC(RegR$optCor, fit$Loadings)
print(C)
```

---

subSet *Subset a data matrix or expression set*

Description

subSet is a convenience function that subsets a data matrix or an ExpressionSet object.

Usage

```
subSet(X, Rf)
```

Arguments

X	A data matrix or an ExpressionSet object.
Rf	A filtered (correlation) matrix (as returned by the RF function).

Details

The subSet convenience function may directly follow usage of the RF in the sense that the latters return-value can be used as the Rf argument. It then subsets a data matrix or an ExpressionSet object to those features retained by the redundancy-filtering. The function returns a subsetted matrix or ExpressionSet (depending on the class of the X argument). The subsetted data can then be used for penalty-parameter selection and regularized correlation matrix estimation provided by the regcor function.

Value

Returns a subsetted data matrix or ExpressionSet.

Note

If argument X is a matrix, the observations are expected to be in the rows and the features are expected to be in the columns.

Author(s)

Carel F.W. Peeters <cf.peeters@vumc.nl>

References

Peeters, C.F.W. *et al.* (2019). Stable prediction with radiomics data. arXiv:1903.11696 [stat.ML].

See Also

regcor

Examples

```
## Generate some (high-dimensional) data
## Get correlation matrix
p = 25
n = 10
set.seed(333)
X = matrix(rnorm(n*p), nrow = n, ncol = p)
colnames(X)[1:25] = letters[1:25]
R <- cor(X)

## Redundancy visualization, at threshold value .9
radioHeat(R, diag = FALSE, threshold = TRUE, threshvalue = .9)

## Redundancy-filtering of correlation matrix
Rfilter <- RF(R, t = .9)
dim(Rfilter)

## Subsetting data
DataSubset <- subSet(X, Rfilter)
dim(DataSubset)
```

Index