Package ‘EFAtools’

July 13, 2020

Title Fast and Flexible Implementations of Exploratory Factor Analysis Tools

Version 0.1.1

Description Provides functions to perform exploratory factor analysis (EFA) procedures and compare their solutions. The goal is to provide state-of-the-art factor retention methods and a high degree of flexibility in the EFA procedures. This way, for example, implementations from R 'psych' and 'SPSS' can be compared. Moreover, functions for Schmid-Leiman transformation and the computation of omegas are provided. To speed up the analyses, some of the iterative procedures, like principal axis factoring (PAF), are implemented in C++.

Depends R (>= 3.6.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports lavaan, psych, crayon, stringr, stats, ggplot2, tibble, magrittr, dplyr, cli, Rcpp, viridisLite, graphics, future.apply, future, GPArotation, checkmate

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, knitr, rmarkdown, microbenchmark

VignetteBuilder knitr

NeedsCompilation yes

Author Markus Steiner [aut, cre],
Silvia Grieder [aut],
William Revelle [ctb],
Max Auerswald [ctb],
Morten Moshagen [ctb],
John Ruscio [ctb],
Brendan Roche [ctb],
Urbano Lorenzo-Seva [ctb],
David Navarro-Gonzalez [ctb]

Maintainer Markus Steiner <markus.d.steiner@gmail.com>

Repository CRAN

Date/Publication 2020-07-13 15:50:18 UTC

R topics documented:

`.compute_vars` *Compute explained variances from loadings*

Description

From unrotated loadings compute the communalities and uniquenesses for total variance. Compute explained variances per factor from rotated loadings (and factor intercorrelations Phi if oblique rotation was used).

Usage

```
.compute_vars(L_unrot, L_rot, Phi = NULL)
```

Arguments

`L_unrot`	matrix. Unrotated factor loadings.
`L_rot`	matrix. Rotated factor loadings.
`Phi`	matrix. Factor intercorrelations. Provide only if oblique rotation is used.

Value

A matrix with sum of squared loadings, proportion explained variance from total variance per factor, same as previous but cumulative, Proportion of explained variance from total explained variance, and same as previous but cumulative.

`.factor_corres` *Perform the iterative PAF procedure*

Description

Function called from within PAF so usually no call to this is needed by the user. Provides a C++ implementation of the PAF procedure

Usage

```
.factor_corres(x, y, thresh = 0.3)
```

Arguments

`x`	numeric matrix. The initial communality estimates.
`y`	numeric matrix. The convergence criterion to use.
`thresh`	numeric. The threshold to classify a pattern coefficient as substantial.

`.numformat` *Format numbers for print method*

Description

Helper function used in the print method for class LOADINGS and SLLOADINGS. Strips the 0 in front of the decimal point of a number if number < 1, only keeps the first `digits` number of digits, and adds an empty space in front of the number if the number is positive. This way all returned strings (except for those > 1, which are exceptions in LOADINGS) have the same number of characters.

Usage

```
.numformat(x, digits = 2, print_zero = FALSE)
```

Arguments

`x`	numeric. Number to be formatted.
`digits`	numeric. Number of digits after the comma to keep.
`print_zero`	logical. Whether, if a number is between]-1, 1[, the zero should be omitted or printed (default is FALSE, i.e. omit zeros).

Value

A formated number

`.paf_iter` *Perform the iterative PAF procedure*

Description

Function called from within PAF so usually no call to this is needed by the user. Provides a C++ implementation of the PAF procedure

Usage

```
.paf_iter(h2, criterion, R, n_fac, abs_eig, crit_type, max_iter)
```

Arguments

`h2`	numeric. The initial communality estimates.
`criterion`	double. The convergence criterion to use.
`R`	matrix. The correlation matrix with the initial communality estimates in the diagonal.
`n_fac`	numeric. The number of factors to extract.
`abs_eig`	logical. Whether absolute eigenvalues should be used to compute the loadings.
`crit_type`	numeric. Whether maximum absolute differences (crit_type = 1), or sum of differences (crit_type = 2) should be used
`max_iter`	numeric. The number of iterations after which to end the procedure if no convergence has been reached by then.

`.parallel_sim` *Parallel analysis on simulated data.*

Description

Function called from within PARALLEL so usually no call to this is needed by the user. Provides a C++ implementation of the PARALLEL simulation procedure

Usage

```
.parallel_sim(n_datasets, n_vars, N, eigen_type)
```

Arguments

`n_datasets`	numeric. Number of datasets with dimensions (N, n_vars) to simulate.
`n_vars`	numeric. Number of variables / indicators in dataset.
`N`	numeric. Number of cases / observations in dataset.
`eigen_type`	numeric. Whether PCA (eigen_type = 1; i.e., leaving diagonal of correlation matrix at 1) or PAF (eigen_type = 2; i.e., setting diagonal of correlation matrix to SMCs).

BARTLETT *Bartlett's test of sphericity*

Description

This function tests whether a correlation matrix is significantly different from an identity matrix (Bartlett, 1951). If the Bartlett's test is not significant, the correlation matrix is not suitable for factor analysis because the variables show too little covariance.

Usage

```
BARTLETT(
  x,
  N = NA,
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall")
)
```

Arguments

x	data.frame or matrix. Dataframe or matrix of raw data or matrix with correlations.
N	numeric. The number of observations. Needs only be specified if a correlation matrix is used.
use	character. Passed to `stats::cor` if raw data is given as input. Default is "pairwise.complete.obs".
cor_method	character. Passed to `stats::cor`. Default is "pearson".

Details

Bartlett (1951) proposed this statistic to determine a correlation matrix' suitability for factor analysis. The statistic is approximately chi square distributed with $df = \frac{p(p-1)}{2}$ and is given by

$$chi^2 = -log(det(R))(N - 1 - (2 * p + 5)/6)$$

where $det(R)$ is the determinant of the correlation matrix, N is the sample size, and p is the number of variables.

This tests requires multivariate normality. If this condition is not met, the Kaiser-Meyer-Olkin criterion (`KMO`) can still be used.

This function was heavily influenced by the `psych::cortest.bartlett` function from the psych package.

The `BARTLETT` function can also be called together with the (`KMO`) function and with factor retention criteria in the `N_FACTORS` function.

Value

A list containing

chisq	The chi square statistic.
p_value	The p value of the chi square statistic.
df	The degrees of freedom for the chi square statistic.
settings	A list of the settings used.

Source

Bartlett, M. S. (1951). The effect of standardization on a Chi-square approximation in factor analysis. Biometrika, 38, 337-344.

See Also

KMO for another measure to determine suitability for factor analysis.

N_FACTORS as a wrapper function for this function, KMO and several factor retention criteria.

Examples

```
BARTLETT(test_models$baseline$cormat, N = 500)
```

CD *Comparison Data*

Description

Factor retention method introduced by Ruscio and Roche (2012). The code was adapted from the CD code by Auerswald and Moshagen (2017) available at https://osf.io/x5cz2/?view_only=d03efba1fd0f4c849a87db82e6705668

Usage

```
CD(
  x,
  n_factors_max = NA,
  N_pop = 10000,
  N_samples = 500,
  alpha = 0.3,
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall"),
  max_iter = 50
)
```

Arguments

`x`	data.frame or matrix. Dataframe or matrix of raw data.
`n_factors_max`	numeric. The maximum number of factors to test against. Larger numbers will increase the duration the procedure takes, but test more possible solutions. If left NA (default) the maximum number of factors for which the model is still over-identified (df > 0) is used.
`N_pop`	numeric. Size of finite populations of comparison data. Default is 10000.
`N_samples`	numeric. Number of samples drawn from each population. Default is 500.
`alpha`	numeric. The alpha level used to test the significance of the improvement added by an additional factor. Default is .30.
`use`	character. Passed to `stats::cor`. Default is "pairwise.complete.obs".
`cor_method`	character. Passed to `stats::cor`. Default is "pearson".
`max_iter`	numeric. The maximum number of iterations to perform after which the iterative PAF procedure is halted. Default is 50.

Details

"Parallel analysis (PA) is an effective stopping rule that compares the eigenvalues of randomly generated data with those for the actual data. PA takes into account sampling error, and at present it is widely considered the best available method. We introduce a variant of PA that goes even further by reproducing the observed correlation matrix rather than generating random data. Comparison data (CD) with known factorial structure are first generated using 1 factor, and then the number of factors is increased until the reproduction of the observed eigenvalues fails to improve significantly" (Ruscio & Roche, 2012, p. 282).

The CD implementation here is based on the code by Ruscio and Roche (2012), but is slightly adapted to increase speed by performing the principal axis factoring using a C++ based function.

The `CD` function can also be called together with other factor retention criteria in the `N_FACTORS` function.

Value

A list of class CD containing

`n_factors`	The number of factors to retain according to comparison data results.
`eigenvalues`	A vector containing the eigenvalues of the entered data.
`RMSE_eigenvalues`	A matrix containing the RMSEs between the eigenvalues of the generated data and those of the entered data.
`settings`	A list of the settings used.

Source

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. Psychological Methods, 24(4), 468–491. https://doi.org/10.1037/met0000200

Ruscio, J., & Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure. Psychological Assessment, 24, 282–292. doi: 10.1037/a0025697

See Also

Other factor retention criteria: EKC, HULL, KGC, PARALLEL, SMT

N_FACTORS as a wrapper function for this and all the above-mentioned factor retention criteria.

Examples

```
# determine n factors of the GRiPS
CD(GRiPS_raw)

# determine n factors of the DOSPERT risk subscale
CD(DOSPERT_raw)
```

COMPARE *Compare two vectors or matrices (communalities or loadings)*

Description

The function takes two objects of the same dimensions containing numeric information (loadings or communalities) and returns a list of class COMPARE containing summary information of the differences of the objects.

Usage

```
COMPARE(
  x,
  y,
  reorder = c("congruence", "names", "none"),
  corres = TRUE,
  thresh = 0.3,
  digits = 4,
  m_red = 0.001,
  range_red = 0.001,
  round_red = 3,
  print_diff = TRUE,
  na.rm = FALSE,
  x_labels = c("x", "y"),
  plot = TRUE,
  plot_red = 0.01
)
```

Arguments

- `x` — matrix, data.frame, or vector. Loadings or communalities of a factor analysis output.
- `y` — matrix, data.frame, or vector. Loadings or communalities of another factor analysis output to compare to x.
- `reorder` — character. Whether and how elements / columns should be reordered. If "congruence" (default), reordering is done according to Tuckers correspondence coefficient, if "names", objects according to their names, if "none", no reordering is done.
- `corres` — logical. Whether factor correspondences should be compared if a matrix is entered.
- `thresh` — numeric. The threshold to classify a pattern coefficient as substantial. Default is .3.
- `digits` — numeric. Number of decimals to print in the output. Default is 4.
- `m_red` — numeric. Number above which the mean and median should be printed in red (i.e., if .001 is used, the mean will be in red if it is larger than .001, otherwise it will be displayed in green.) Default is .001.
- `range_red` — numeric. Number above which the min and max should be printed in red (i.e., if .001 is used, min and max will be in red if the max is larger than .001, otherwise it will be displayed in green. Default is .001). Note that the color of min also depends on max, that is min will be displayed in the same color as max.
- `round_red` — numeric. Number above which the max decimals to round to where all corresponding elements of x and y are still equal are displayed in red (i.e., if 3 is used, the number will be in red if it is smaller than 3, otherwise it will be displayed in green). Default is 3.
- `print_diff` — logical. Whether the difference vector or matrix should be printed or not. Default is TRUE.
- `na.rm` — logical. Whether NAs should be removed in the mean, median, min, and max functions. Default is FALSE.
- `x_labels` — character. A vector of length two containing identifying labels for the two objects x and y that will be compared. These will be used as labels on the x-axis of the plot. Default is "x" and "y".
- `plot` — logical. If TRUE (default), a plot illustrating the differences will be shown.
- `plot_red` — numeric. Threshold above which to plot the absolute differences in red. Default is .001.

Value

A list of class COMPARE containing summary statistics on the differences of x and y.

- `diff` — The vector or matrix containing the differences between x and y.
- `mean_abs_diff` — The mean absolute difference between x and y.
- `median_abs_diff` — The median absolute difference between x and y.

`min_abs_diff` The minimum absolute difference between x and y.

`max_abs_diff` The maximum absolute difference between x and y.

`max_dec` The maximum number of decimals to which a comparison makes sense. For example, if x contains only values up to the third decimals, and y is a normal double, max_dec will be three.

`are_equal` The maximal number of decimals to which all elements of x and y are equal.

`diff_corres` The number of differing variable-to-factor correspondences between x and y, when only the highest loading is considered.

`diff_corres_cross`
The number of differing variable-to-factor correspondences between x and y when all loadings >= `thresh` are considered.

`g` The root mean squared distance (RMSE) between x and y.

`settings` List of the settings used.

Examples

```
# A type SPSS EFA to mimick the SPSS implementation
EFA_SPSS_6 <- EFA(test_models$case_11b$cormat, n_factors = 6, type = "SPSS")

# A type psych EFA to mimick the psych::fa() implementation
EFA_psych_6 <- EFA(test_models$case_11b$cormat, n_factors = 6, type = "psych")

# compare the two
COMPARE(EFA_SPSS_6$unrot_loadings, EFA_psych_6$unrot_loadings,
        x_labels = c("SPSS", "psych"))
```

DOSPERT *DOSPERT*

Description

A list containing the the bivariate correlations (cormat) of the 40 items of the Domain Specific Risk Taking Scale (DOSPERT; Weber, Blais, & Betz, 2002) and the sample size (N) based on the publicly available dataset at (https://osf.io/rce7g) of the Basel-Berlin Risk Study (Frey et al., 2017). The items measure risk-taking propensity on six different domains: social, recreational, gambling, health/ safety, investment, and ethical.

Usage

```
DOSPERT
```

Format

An object of class `list` of length 2.

Source

Weber, E. U., Blais, A.-R., & Betz, N. E. (2002). A domain specific risk-attitude scale: Measuring risk perceptions and risk behaviors. Journal of Behavioral Decision Making, 15(4), 263–290. doi: 10.1002/bdm.414

Frey, R., Pedroni, A., Mata, R., Rieskamp, J., & Hertwig, R. (2017). Risk preference shares the psychometric structure of major psychological traits. Science Advances, 3, e1701381.

https://osf.io/rce7g

DOSPERT_raw — *DOSPERT_raw*

Description

A data.frame containing responses to the risk subscale of the Domain Specific Risk Taking Scale (DOSPERT; Weber, Blais, & Betz, 2002) based on the publicly available dataset (at https://osf.io/pjt57/) by Frey, Duncan, and Weber (2020). The items measure risk-taking propensity on six different domains: social, recreational, gambling, health/ safety, investment, and ethical.

Usage

```
DOSPERT_raw
```

Format

An object of class `data.frame` with 3123 rows and 30 columns.

Source

Blais, A.-R., & Weber, E. U. (2002). A domain-specific risk-taking (DOSPERT) scale for adult populations. Judgment and Decision Making, 15(4), 263–290. doi: 10.1002/bdm.414

Frey, R., Duncan, S. M., & Weber, E. U. (2020). Towards a typology of risk preference: Four risk profiles describe two thirds of individuals in a large sample of the U.S. population. PsyArXiv Preprint. doi:10.31234/osf.io/yjwr9

EFA — *Exploratory factor analysis (EFA)*

Description

This function does an EFA with either `PAF`, `ML`, or `ULS` with or without subsequent rotation. All arguments with default value `NULL` can be left to default if `type` is set to one of "EFAtools", "SPSS", or "psych". The respective specifications are then handled according to the specified type (see details). For all rotations except varimax and promax, the `GPArotation` package is needed.

Usage

```
EFA(
  x,
  n_factors,
  N = NA,
  method = c("PAF", "ML", "ULS"),
  rotation = c("none", "varimax", "equamax", "quartimax", "geominT", "bentlerT",
   "bifactorT", "promax", "oblimin", "quartimin", "simplimax", "bentlerQ", "geominQ",
    "bifactorQ"),
  type = c("EFAtools", "psych", "SPSS", "none"),
  max_iter = NULL,
  init_comm = NULL,
  criterion = NULL,
  criterion_type = NULL,
  abs_eigen = NULL,
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  k = NULL,
  kaiser = TRUE,
  P_type = NULL,
  precision = NULL,
  order_type = NULL,
  start_method = c("factanal", "psych"),
  cor_method = c("pearson", "spearman", "kendall"),
  ...
)
```

Arguments

`x`	data.frame or matrix. Dataframe or matrix of raw data or matrix with correlations. If raw data is entered, the correlation matrix is found from the data.
`n_factors`	numeric. Number of factors to extract.
`N`	numeric. The number of observations. Needs only be specified if a correlation matrix is used. If input is a correlation matrix and `N` = NA (default), not all fit indices can be computed. See `psych::factor.stats` for details.
`method`	character. One of "PAF", "ML", or "ULS" to use principal axis factoring, maximum likelihood, or unweighted least squares (also called minres), respectively, to fit the EFA.
`rotation`	character. Either perform no rotation ("none"; default), an orthogonal rotation ("varimax", "equamax", "quartimax", "geominT", "bentlerT", or "bifactorT"), or an oblique rotation ("promax", "oblimin", "quartimin", "simplimax", "bentlerQ", "geominQ", or "bifactorQ").
`type`	character. If one of "EFAtools" (default), "psych", or "SPSS" is used, and the following arguments with default NULL are left with NULL, these implementations are executed according to the respective program ("psych" and "SPSS") or according to the best solution found in Grieder & Steiner (2020; "EFAtools"). Individual properties can be adapted using one of the three types and specifying

some of the following arguments. If set to "none" additional arguments must be specified depending on the method and rotation used (see details).

max_iter
: numeric. The maximum number of iterations to perform after which the iterative PAF procedure is halted with a warning. If type is one of "EFAtools", "SPSS", or "psych", this is automatically specified if max_iter is left to be NULL, but can be overridden by entering a number. Default is NULL.

init_comm
: character. The method to estimate the initial communalities in PAF. "smc" will use squared multiple correlations, "mac" will use maximum absolute correlations, "unity" will use 1s (see details). Default is NULL.

criterion
: numeric. The convergence criterion used for PAF. If the change in communalities from one iteration to the next is smaller than this criterion the solution is accepted and the procedure ends. Default is NULL.

criterion_type
: character. Type of convergence criterion used for PAF. "max_individual" selects the maximum change in any of the communalities from one iteration to the next and tests it against the specified criterion. This is also used by SPSS. "sums" takes the difference of the sum of all communalities in one iteration and the sum of all communalities in the next iteration and tests this against the criterion. This procedure is used by the psych::fa function. Default is NULL.

abs_eigen
: logical. Which algorithm to use in the PAF iterations. If FALSE, the loadings are computed from the eigenvalues. This is also used by the psych::fa function. If TRUE the loadings are computed with the absolute eigenvalues as done by SPSS. Default is NULL.

use
: character. Passed to stats::cor if raw data is given as input. Default is "pairwise.complete.obs".

k
: numeric. Either the power used for computing the target matrix P in the promax rotation or the number of 'close to zero loadings' for the simplimax rotation (see GPArotation::GPFoblq). If left to NULL (default), the value for promax depends on the specified type. For simplimax, nrow(L), where L is the matrix of unrotated loadings, is used by default.

kaiser
: logical. If TRUE, kaiser normalization is performed before the specified rotation. Default is TRUE.

P_type
: character. This specifies how the target matrix P is computed in promax rotation. If "unnorm" it will use the unnormalized target matrix as originally done in Hendrickson and White (1964). This is also used in the psych and stats packages. If "norm" it will use the normalized target matrix as used in SPSS. Default is NULL.

precision
: numeric. The tolerance for stopping in the rotation procecdure. This is passed to the "eps" argument of the stats::varimax and the GPArotation functions. If left NULL (default), the precision is set according to the specified type for varimax and promax rotation or is set to 10^-5 for all other rotations.

order_type
: character. How to order the factors. "eigen" will reorder the factors according to the largest to lowest eigenvalues of the matrix of rotated loadings. "ss_factors" will reorder the factors according to descending sum of squared factor loadings per factor. Default is NULL.

start_method — character. How to specify the starting values for the optimization procedure for ML. Default is "factanal" which takes the starting values specified in the stats::factanal function. "psych" takes the starting values specified in psych::fa. Solutions are very similar.

cor_method — character. Passed to `stats::cor`. Default is "pearson".

... — Additional arguments passed to rotation functions from the `GPArotation` package (e.g., `maxit` for maximum number of iterations).

Details

There are two main ways to use this function. The easiest way is to use it with a specified `type` (see above), which sets most of the other arguments accordingly. Another way is to use it more flexibly by explicitly specifying all arguments used and set `type` to "none" (see examples). A mix of the two can also be done by specifying a `type` as well as additional arguments. However, this will throw warnings to avoid unintentional deviations from the implementations according to the specified `type`.

The `type` argument is evaluated for PAF and for all rotations (mainly important for the varimax and promax rotations). The type-specific settings for these functions are detailed below.

For PAF, the values of `init_comm`, `criterion`, `criterion_type`, and `abs_eigen` depend on the `type` argument.

`type = "EFAtools"` will use the following argument specification: `init_comm = "mac", criterion = .001, criterion_type = "sums", abs_eigen = TRUE`.

`type = "psych"` will use the following argument specification: `init_comm = "smc", criterion = .001, criterion_type = "sums", abs_eigen = FALSE`.

`type = "SPSS"` will use the following argument specification: `init_comm = "smc", criterion = .001, criterion_type = "max_individual", abs_eigen = TRUE`.

If SMCs fail, SPSS takes "mac". However, as SPSS takes absolute eigenvalues, this is hardly ever the case. Psych, on the other hand, takes "unity" if SMCs fail. The EFAtools type setting combination was the best in terms of accuracy and number of Heywood cases compared to all the other setting combinations tested in simulation studies in Grieder & Steiner (2020), which is why this type is used as a default here.

For varimax, the values of `precision` and `order_type` depend on the `type` argument.

`type = "EFAtools"` will use the following argument specification: `precision = 1e-5, order_type = "eigen"`.

`type = "psych"` will use the following argument specification: `precision = 1e-5, order_type = "eigen"`.

`type = "SPSS"` will use the following argument specification: `precision = 1e-10, order_type = "ss_factors"`.

For promax, the values of `P_type`, `precision`, `order_type`, and `k` depend on the `type` argument.

`type = "EFAtools"` will use the following argument specification: `P_type = "unnorm", precision = 1e-5, order_type = "eigen", k = 3`.

`type = "psych"` will use the following argument specification: `P_type = "unnorm", precision = 1e-5, order_type = "eigen", k = 4`.

type = "SPSS" will use the following argument specification: P_type = "norm",precision = 1e-10,order_type = "ss_factors",k = 4.

The P_type argument can take two values, "unnorm" and "norm". It controls which formula is used to compute the target matrix P in the promax rotation. "unnorm" uses the formula from Hendrickson and White (1964), specifically: P = abs(A^(k + 1)) / A, where A is the unnormalized matrix containing varimax rotated loadings. "SPSS" uses the normalized varimax rotated loadings. Specifically it used the following formula, which can be found in the SPSS 23 Algorithms manual: P = abs(A / sqrt(rowSums(A^2))) ^(k + 1) * (sqrt(rowSums(A^2)) / A). As for PAF, the EFAtools type setting combination for promax was the best compared to the other setting combinations tested in simulation studies in Grieder & Steiner (2020).

For all other rotations except varimax and promax, the type argument only controls the order_type argument with the same values as stated above for the varimax and promax rotations. For these other rotations, the GPArotation package is needed. Additional arguments can also be specified and will be passed to the respective GPArotation function (e.g., maxit to change the maximum number of iterations for the rotation procedure).

The type argument has no effect on ULS and ML. For ULS, no additional arguments are needed. For ML, an additional argument start_method is needed to determine the starting values for the optimization procedure. Default for this argument is "factanal" which takes the starting values specified in the stats::factanal function.

Value

A list of class EFA containing (a subset of) the following:

orig_R	Original correlation matrix.
h2_init	Initial communality estimates from PAF.
h2	Final communality estimates from the unrotated solution.
orig_eigen	Eigen values of the original correlation matrix.
init_eigen	Initial eigenvalues, obtained from the correlation matrix with the initial communality estimates as diagonal in PAF.
final_eigen	Eigenvalues obtained from the correlation matrix with the final communality estimates as diagonal.
iter	The number of iterations needed for convergence.
convergence	Integer code for convergence as returned by stats:optim (only for ML and ULS). 0 indicates successful completion.
unrot_loadings	Loading matrix containing the final unrotated loadings.
vars_accounted	Matrix of explained variances and sums of squared loadings. Based on the unrotated loadings.
fit_indices	For ML and ULS: Fit indices derived from the unrotated factor loadings: Chi Square, degrees of freedom (df), Comparative Fit Index (CFI), Root Mean Square Error of Approximation (RMSEA), including its 90% confidence interval, and the common part accounted for (CAF) index as proposed by Lorenzo-Seva, Timmerman, & Kiers (2011). For PAF, only the CAF and dfs are returned.
rot_loadings	Loading matrix containing the final rotated loadings (pattern matrix).

`Phi` The factor intercorrelations (only for oblique rotations).

`Structure` The structure matrix (only for oblique rotations).

`rotmat` The rotation matrix.

`vars_accounted_rot`
Matrix of explained variances and sums of squared loadings. Based on rotated loadings and, for oblique rotations, the factor intercorrelations.

`settings` A list of the settings used.

Source

Grieder, S., & Steiner, M.D. (2020). Algorithmic Jingle Jungle: A Comparison of Implementations of Principal Axis Factoring and Promax Rotation in R and SPSS. Manuscript in Preparation.

Hendrickson, A. E., & White, P. O. (1964). Promax: A quick method for rotation to oblique simple structure. British Journal of Statistical Psychology, 17 , 65–70. doi: 10.1111/j.2044-8317.1964.tb00244.x

Lorenzo-Seva, U., Timmerman, M. E., & Kiers, H. A. L. (2011). The Hull Method for Selecting the Number of Common Factors, Multivariate Behavioral Research, 46, 340-364, doi: 10.1080/00273171.2011.564527

Examples

```
# A type EFAtools (as presented in Steiner and Grieder, 2020) EFA
EFAtools_PAF <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                    type = "EFAtools", method = "PAF", rotation = "none")

# A type SPSS EFA to mimick the SPSS implementation (this will throw a warning,
# see below)
SPSS_PAF <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                type = "SPSS", method = "PAF", rotation = "none")

# A type psych EFA to mimick the psych::fa() implementation
psych_PAF <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                 type = "psych", method = "PAF", rotation = "none")

# Use ML instead of PAF with type EFAtools
EFAtools_ML <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                   type = "EFAtools", method = "ML", rotation = "none")

# Use oblimin rotation instead of no rotation with type EFAtools
EFAtools_oblim <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                      type = "EFAtools", method = "PAF", rotation = "oblimin")

# Do a PAF without rotation without specifying a type, so the arguments
# can be flexibly specified (this is only recommended if you know what your
# doing)
PAF_none <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                type = "none", method = "PAF", rotation = "none",
                max_iter = 500, init_comm = "mac", criterion = 1e-4,
                criterion_type = "sums", abs_eigen = FALSE)
```

```
# Add a promax rotation
PAF_pro <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
               type = "none", method = "PAF", rotation = "promax",
               max_iter = 500, init_comm = "mac", criterion = 1e-4,
               criterion_type = "sums", abs_eigen = FALSE, k = 3,
               P_type = "unnorm", precision= 1e-5, order_type = "eigen")
```

EKC *Empirical Kaiser Criterion*

Description

The empirical Kaiser criterion incorporates random sampling variations of the eigenvalues from the Kaiser-Guttman criterion (KGC; see Auerswald & Moshagen , 2019; Braeken & van Assen, 2017). The code is based on Auerswald and Moshagen (2019).

Usage

```
EKC(
  x,
  N = NA,
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall")
)
```

Arguments

x	data.frame or matrix. data.frame or matrix of raw data or matrix with correlations.
N	numeric. The number of observations. Only needed if x is a correlation matrix.
use	character. Passed to `stats::cor` if raw data is given as input. Default is `"pairwise.complete.obs"`.
cor_method	character. Passed to `stats::cor`. Default is `"pearson"`.

Details

The Kaiser-Guttman criterion was defined with the intend that a factor should only be extracted if it explains at least as much variance as a single factor (see KGC). However, this only applies to population-level correlation matrices. Due to sampling variation, the KGC strongly overestimates the number of factors to retrieve (e.g., Zwick & Velicer, 1986). To account for this and to introduce a factor retention method that performs well with small number of indicators and correlated factors (cases where the performance of parallel analysis, see PARALLEL, is known to deteriorate) Braeken and van Assen (2017) introduced the empirical Kaiser criterion in which a series of reference eigenvalues is created as a function of the variables-to-sample-size ratio and the observed eigenvalues.

Braeken and van Assen (2017) showed that "(a) EKC performs about as well as parallel analysis for data arising from the null, 1-factor, or orthogonal factors model; and (b) clearly outperforms parallel analysis for the specific case of oblique factors, particularly whenever factor intercorrelation is moderate to high and the number of variables per factor is small, which is characteristic of many applications these days" (p.463-464).

The `EKC` function can also be called together with other factor retention criteria in the `N_FACTORS` function.

Value

A list of class EKC containing

`eigenvalues`	A vector containing the eigenvalues found on the correlation matrix of the entered data.
`n_factors`	The number of factors to retain according to the empirical Kaiser criterion.
`references`	The reference eigenvalues.
`settings`	A list with the settings used.

Source

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. Psychological Methods, 24(4), 468–491. https://doi.org/10.1037/met0000200

Braeken, J., & van Assen, M. A. (2017). An empirical Kaiser criterion. Psychological Methods, 22, 450 – 466. http://dx.doi.org/10.1037/ met0000074

Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. Psychological Bulletin, 99, 432–442. http://dx.doi.org/10.1037/0033-2909.99.3.432

See Also

Other factor retention criteria: `CD`, `HULL`, `KGC`, `PARALLEL`, `SMT`

`N_FACTORS` as a wrapper function for this and all the above-mentioned factor retention criteria.

Examples

```
EKC(test_models$baseline$cormat, N = 500)
```

GRiPS_raw *GRiPS_raw*

Description

A data.frame containing responses to the General Risk Propensity Scale (GRiPS, Zhang, Highhouse & Nye, 2018) of 810 participants of Study 1 of Steiner and Frey (2020). The original data can be accessed via https://osf.io/kxp8t/.

Usage

```
GRiPS_raw
```

Format

An object of class `data.frame` with 810 rows and 8 columns.

Source

Zhang, D. C., Highhouse, S., & Nye, C. D. (2018). Development and validation of the general risk propensity scale (GRiPS).Journal of Behavioral Decision Making, 32, 152–167. doi: 10.1002/bdm.2102

Steiner, M., & Frey, R. (2020). Representative design in psychological assessment: A case study using the Balloon Analogue Risk Task (BART). PsyArXiv Preprint. doi:10.31234/osf.io/dg4ks

HULL *Hull method for determining the number of factors to retain*

Description

Implementation of the Hull method suggested by Lorenzo-Seva, Timmerman, and Kiers (2011), with an extension to principal axis factoring. See details for parallelization.

Usage

```
HULL(
  x,
  N = NA,
  n_fac_theor = NA,
  method = c("PAF", "ULS", "ML"),
  gof = c("CAF", "CFI", "RMSEA"),
  eigen_type = c("SMC", "PCA", "EFA"),
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall"),
```

```
  n_datasets = 1000,
  percent = 95,
  decision_rule = c("means", "percentile", "crawford"),
  n_factors = 1,
  ...
)
```

Arguments

x	matrix or data.frame. Dataframe or matrix of raw data or matrix with correlations.
N	numeric. Number of cases in the data. This is passed to PARALLEL. Only has to be specified if x is a correlation matrix, otherwise it is determined based on the dimensions of x.
n_fac_theor	numeric. Theoretical number of factors to retain. The maximum of this number and the number of factors suggested by PARALLEL plus one will be used in the Hull method.
method	character. The estimation method to use. One of "PAF", "ULS", or "ML", for principal axis factoring, unweighted least squares, and maximum likelihood, respectively.
gof	character. The goodness of fit index to use. Either "CAF", "CFI", or "RMSEA", or any combination of them. If method = "PAF" is used, only the CAF can be used as goodness of fit index. For details on the CAF, see Lorenzo-Seva, Timmerman, and Kiers (2011).
eigen_type	character. On what the eigenvalues should be found in the parallel analysis.. Can be one of "SMC", "PCA", or "EFA". If using "SMC" (default), the diagonal of the correlation matrices is replaced by the squared multiple correlations (SMCs) of the indicators. If using "PCA", the diagonal values of the correlation matrices are left to be 1. If using "EFA", eigenvalues are found on the correlation matrices with the final communalities of an EFA solution as diagonal. This is passed to PARALLEL.
use	character. Passed to stats::cor if raw data is given as input. Default is "pairwise.complete.obs".
cor_method	character. Passed to stats::cor. Default is "pearson".
n_datasets	numeric. The number of datasets to simulate. Default is 1000. This is passed to PARALLEL.
percent	numeric. A vector of percentiles to take the simulated eigenvalues from. Default is 95. This is passed to PARALLEL.
decision_rule	character. Which rule to use to determine the number of factors to retain. Default is "means", which will use the average simulated eigenvalues. "percentile", uses the percentiles specified in percent. "crawford" uses the 95th percentile for the first factor and the mean afterwards (based on Crawford et al, 2010). This is passed to PARALLEL.
n_factors	numeric. Number of factors to extract if "EFA" is included in eigen_type. Default is 1. This is passed to PARALLEL.
...	Further arguments passed to EFA, also in PARALLEL.

Details

The Hull method aims to find a model with an optimal balance between model fit and number of parameters. That is, it aims to retrieve only major factors (Lorenzo-Seva, Timmerman, & Kiers, 2011). To this end, it performs the following steps (Lorenzo-Seva, Timmerman, & Kiers, 2011, p.351):

1. It performs parallel analysis and adds one to the identified number of factors (this number is denoted *J*). *J* is taken as an upper bound of the number of factors to retain in the hull method. Alternatively, a theoretical number of factors can be entered. In this case *J* will be set to whichever of these two numbers (from parallel analysis or based on theory) is higher.
2. For all 0 to *J* factors, the goodness-of-fit (one of *CAF*, *RMSEA*, or *CFI*) and the degrees of freedom (*df*) are computed.
3. The solutions are ordered according to their *df*.
4. Solutions that are not on the boundary of the convex hull are eliminated (see Lorenzo-Seva, Timmerman, & Kiers, 2011, for details).
5. All the triplets of adjacent solutions are considered consecutively. The middle solution is excluded if its point is below or on the line connecting its neighbors in a plot of the goodness-of-fit versus the degrees of freedom.
6. Step 5 is repeated until no solution can be excluded.
7. The *st* values of the “hull” solutions are determined.
8. The solution with the highest *st* value is selected.

The PARALLEL function and the principal axis factoring of the different number of factors can be parallelized using the future framework, by calling the future::plan function. The examples provide example code on how to enable parallel processing.

Note that if `gof = "RMSEA"` is used, 1 - RMSEA is actually used to compare the different solutions. Thus, the threshold of .05 is then .95. This is necessary due to how the heuristic to locate the elbow of the hull works.

The ML estimation method uses the stats::factanal starting values. See also the EFA documentation.

The `HULL` function can also be called together with other factor retention criteria in the `N_FACTORS` function.

Value

A list of class HULL containing the following objects

`n_fac_CAF`	The number of factors to retain according to the Hull method with the CAF.
`n_fac_CFI`	The number of factors to retain according to the Hull method with the CFI.
`n_fac_RMSEA`	The number of factors to retain according to the Hull method with the RMSEA.
`solutions_CAF`	A matrix containing the CAFs, degrees of freedom, and for the factors lying on the hull, the st values of the hull solution (see Lorenzo-Seva, Timmerman, and Kiers 2011 for details).
`solutions_CFI`	A matrix containing the CFIs, degrees of freedom, and for the factors lying on the hull, the st values of the hull solution (see Lorenzo-Seva, Timmerman, and Kiers 2011 for details).

solutions_RMSEA
: A matrix containing the RMSEAs, degrees of freedom, and for the factors lying on the hull, the st values of the hull solution (see Lorenzo-Seva, Timmerman, and Kiers 2011 for details).

n_fac_max
: The upper bound J of the number of factors to extract (see details).

settings
: A list of the settings used.

Source

Lorenzo-Seva, U., Timmerman, M. E., & Kiers, H. A. (2011). The Hull method for selecting the number of common factors. Multivariate Behavioral Research, 46(2), 340-364.

See Also

Other factor retention criteria: CD, EKC, KGC, PARALLEL, SMT

N_FACTORS as a wrapper function for this and all the above-mentioned factor retention criteria.

Examples

```
# using PAF (this will throw a warning if gof is not specified manually
# and CAF will be used automatically)
HULL(test_models$baseline$cormat, N = 500, gof = "CAF")

# using ML with all available fit indices (CAF, CFI, and RMSEA)
HULL(test_models$baseline$cormat, N = 500, method = "ML")

# using ULS with only RMSEA
HULL(test_models$baseline$cormat, N = 500, method = "ULS", gof = "RMSEA")


## Not run:
# using parallel processing (Note: plans can be adapted, see the future
# package for details)
future::plan(future::multisession)
HULL(test_models$baseline$cormat, N = 500, gof = "CAF")

## End(Not run)
```

IDS2_R *Intelligence subtests from the Intelligence and Development Scales–2*

Description

A matrix containing the bivariate correlations of the 14 intelligence subtests from the Intelligence and Development Scales–2 (IDS-2; Grob & Hagmann-von Arx, 2018), an intelligence and development test battery for children and adolescents aged 5 to 20 years, for the standardization and validation sample (N = 1,991). Details can be found in Grieder & Grob (2019).

Usage

```
IDS2_R
```

Format

A 14 x 14 matrix of bivariate correlations

GS (numeric) - Geometric shapes.

PL (numeric) - Plates.

TC (numeric) - Two characteristics.

CB (numeric) - Crossing out boxes.

NL (numeric) - Numbers / letters.

NLM (numeric) - Numbers / letter mixed.

GF (numeric) - Geometric figures.

RGF (numeric) - Rotated geometric figures.

CM (numeric) - Completing matrices.

EP (numeric) - Excluding pictures.

CA (numeric) - Categories.

OP (numeric) - Opposites.

RS (numeric) - Retelling a story.

DP (numeric) - Describing pictures.

Source

Grieder, S., & Grob, A. (2019). Exploratory factor analyses of the intelligence and development scales–2: Implications for theory and practice. Assessment. Advance online publication. doi:10.1177/10731911198450

Grob, A., & Hagmann-von Arx, P. (2018). Intelligence and Development Scales–2 (IDS-2). Intelligenz- und Entwicklungsskalen für Kinder und Jugendliche. [Intelligence and Development Scales for Children and Adolescents.]. Bern, Switzerland: Hogrefe.

`KGC` *Kaiser-Guttman Criterion*

Description

Probably the most popular factor retention criterion. Kaiser and Guttman suggested to retain as many factors as there are sample eigenvalues greater than 1. This is why the criterion is also known as eigenvalues-greater-than-one rule.

Usage

```
KGC(
  x,
  eigen_type = c("PCA", "SMC", "EFA"),
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall"),
  n_factors = 1,
  ...
)
```

Arguments

`x`	data.frame or matrix. Dataframe or matrix of raw data or matrix with correlations.
`eigen_type`	character. On what the eigenvalues should be found. Can be either "PCA", "SMC", or "EFA", or some combination of them. If using "PCA", the diagonal values of the correlation matrices are left to be 1. If using "SMC", the diagonal of the correlation matrices is replaced by the squared multiple correlations (SMCs) of the indicators. If using "EFA", eigenvalues are found on the correlation matrices with the final communalities of an exploratory factor analysis solution (default is principal axis factoring extracting 1 factor) as diagonal.
`use`	character. Passed to `stats::cor` if raw data is given as input. Default is "pairwise.complete.obs".
`cor_method`	character. Passed to `stats::cor`. Default is "pearson".
`n_factors`	numeric. Number of factors to extract if "EFA" is included in `eigen_type`. Default is 1.
`...`	Additional arguments passed to `EFA`. For example, to change the extraction method (PAF is default).

Details

Originally, the Kaiser-Guttman criterion was intended for the use with prinicpal components, hence with eigenvalues derived from the original correlation matrix. This can be done here by setting `eigen_type` to "PCA". However, it is well-known that this criterion is often inaccurate and that it tends to overestimate the number of factors, especially for unidimensional or orthogonal factor structures (e.g., Zwick & Velicer, 1986).

The criterion's inaccuracy in these cases is somewhat addressed if it is applied on the correlation matrix with communalities in the diagonal, either initial communalities estimated from SMCs (done setting `eigen_type` to "SMC") or final communality estimates from an EFA (done setting `eigen_type` to "EFA"; see Auerswald & Moshagen, 2019). However, although this variant of the KGC is more accurate in some cases compared to the traditional KGC, it is at the same time less accurate than the PCA-variant in other cases, and it is still often less accurate than other factor retention methods, for example parallel analysis (`PARALLEL`), the Hull method `HULL`, or sequential chi^2 model tests (`SMT`; see Auerswald & Moshagen, 2019).

The `KGC` function can also be called together with other factor retention criteria in the `N_FACTORS` function.

Value

A list of class KGC containing

eigen_PCA	A vector containing the eigenvalues found with PCA.
eigen_SMC	A vector containing the eigenvalues found with SMCs.
eigen_EFA	A vector containing the eigenvalues found with EFA.
n_fac_PCA	The number of factors to retain according to the Kaiser- Guttmann criterion with PCA eigenvalues type.
n_fac_SMC	The number of factors to retain according to the Kaiser- Guttmann criterion with SMC eigenvalues type.
n_fac_EFA	The number of factors to retain according to the Kaiser- Guttmann criterion with EFA eigenvalues type.
settings	A list of the settings used.

Source

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. Psychological Methods, 24(4), 468–491. https://doi.org/10.1037/met0000200

Guttman, L. (1954). Some necessary conditions for common-factor analysis. Psychometrika, 19, 149 –161. http://dx.doi.org/10.1007/BF02289162

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. Educational and Psychological Measurement, 20, 141–151. http://dx.doi.org/10.1177/001316446002000116

Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. Psychological Bulletin, 99, 432–442. http://dx.doi.org/10.1037/0033-2909.99.3.432

See Also

Other factor retention criteria: CD, EKC, HULL, PARALLEL, SMT

N_FACTORS as a wrapper function for this and all the above-mentioned factor retention criteria.

Examples

```
KGC(test_models$baseline$cormat, eigen_type = c("PCA", "SMC"))
```

KMO *Kaiser-Meyer-Olkin criterion*

Description

This function computes the Kaiser-Meyer-Olkin (KMO) criterion overall and for each variable in a correlation matrix. The KMO represents the degree to which each observed variable is predicted by the other variables in the dataset and with this indicates the suitability for factor analysis.

Usage

```
KMO(
  x,
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall")
)
```

Arguments

x	data.frame or matrix. Dataframe or matrix of raw data or matrix with correlations.
use	character. Passed to stats::cor if raw data is given as input. Default is "pairwise.complete.obs".
cor_method	character. Passed to stats::cor. Default is "pearson".

Details

Kaiser (1970) proposed this index, originally called measure of sampling adequacy (MSA), that indicates how near the inverted correlation matrix R^{-1} is to a diagonal matrix S to determine a given correlation matrix's (R) suitability for factor analysis. The index is

$$KMO = \frac{\sum\limits_{i<j}\sum r_{ij}^2}{\sum\limits_{i<j}\sum r_{ij}^2 + \sum\limits_{i<j}\sum q_{ij}^2}$$

with $Q = SR^{-1}S$ and S = $(diagR^{-1})^{-1/2}$ where $\sum\limits_{i<j}\sum r_{ij}^2$ is the sum of squares of the upper off-diagonal elements of R and $\sum\limits_{i<j}\sum q_{ij}^2$ is the sum of squares of the upper off-diagonal elements of Q (see also Cureton & D'Augustino, 1983).

So KMO varies between 0 and 1, with larger values indicating higher suitability for factor analysis. Kaiser and Rice (1974) suggest that KMO should at least exceed .50 for a correlation matrix to be suitable for factor analysis.

This function was heavily influenced by the psych::KMO function.

See also BARTLETT for another test of suitability for factor analysis.

The KMO function can also be called together with the BARTLETT function and with factor retention criteria in the N_FACTORS function.

Value

A list containing

KMO	Overall KMO.
KMO_i	KMO for each variable.
settings	A list of the settings used.

Source

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 Psychological Measurement, 34, 111-117.

Cureton, E. E. & D'Augustino, R. B. (1983). Factor analysis: An applied approach. Hillsdale, N.J.: Lawrence Erlbaum Associates, Inc.

See Also

BARTLETT for another measure to determine suitability for factor analysis.

N_FACTORS as a wrapper function for this function, BARTLETT and several factor retention criteria.

Examples

```
KMO(test_models$baseline$cormat)
```

N_FACTORS *Various Factor Retention Criteria*

Description

Among the most important decisions for an exploratory factor analysis (EFA) is the choice of the number of factors to retain. Several factor retention criteria have been developed for this. With this function, various factor retention criteria can be performed simultaneously. Additionally, the data can be checked for their suitability for factor analysis.

Usage

```
N_FACTORS(
  x,
  criteria = c("CD", "EKC", "HULL", "KGC", "PARALLEL", "SMT"),
  suitability = TRUE,
  N = NA,
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall"),
  n_factors_max = NA,
  N_pop = 10000,
```

```
  N_samples = 500,
  alpha = 0.3,
  max_iter_CD = 50,
  n_fac_theor = NA,
  method = c("PAF", "ULS", "ML"),
  gof = c("CAF", "CFI", "RMSEA"),
  eigen_type_HULL = c("SMC", "PCA", "EFA"),
  eigen_type_KGC_PA = c("PCA", "SMC", "EFA"),
  n_factors = 1,
  n_datasets = 1000,
  percent = 95,
  decision_rule = c("means", "percentile", "crawford"),
  ...
)
```

Arguments

x	data.frame or matrix. Dataframe or matrix of raw data or matrix with correlations. If "CD" is included as a criterion, x must be raw data.
criteria	character. A vector with the factor retention methods to perform. Possible inputs are: "CD", "EKC", "HULL", "KGC", "PARALLEL", and "SMT" (see details). By default, all factor retention methods are performed.
suitability	logical. Whether the data should be checked for suitability for factor analysis using the Bartlett's test of sphericity and the Kaiser-Guttmann criterion (see details). Default is TRUE.
N	numeric. The number of observations. Only needed if x is a correlation matrix.
use	character. Passed to stats::cor if raw data is given as input. Default is "pairwise.complete.obs".
cor_method	character. Passed to stats::cor Default is "pearson".
n_factors_max	numeric. Passed to CD.The maximum number of factors to test against. Larger numbers will increase the duration the procedure takes, but test more possible solutions. Maximum possible is number of variables / 2. Default is NA. If not specified, number of variables / 2 is used.
N_pop	numeric. Passed to CD. Size of finite populations of comparison data. Default is 10000.
N_samples	numeric. Passed to CD. Number of samples drawn from each population. Default is 500.
alpha	numeric. Passed to CD. The alpha level used to test the significance of the improvement added by an additional factor. Default is .30.
max_iter_CD	numeric. Passed to CD. The maximum number of iterations to perform after which the iterative PAF procedure is halted. Default is 50.
n_fac_theor	numeric. Passed to HULL. Theoretical number of factors to retain. The maximum of this number and the number of factors suggested by PARALLEL plus one will be used in the Hull method.

method
: character. Passed to EFA in HULL, KGC, and PARALLEL. The estimation method to use. One of "PAF", "ULS", or "ML", for principal axis factoring, unweighted least squares, and maximum likelihood, respectively.

gof
: character. Passed to HULL. The goodness of fit index to use. Either "CAF", "CFI", or "RMSEA", or any combination of them. If method = "PAF" is used, only the CAF can be used as goodness of fit index. For details on the CAF, see Lorenzo-Seva, Timmerman, and Kiers (2011).

eigen_type_HULL
: character. Passed to PARALLEL in HULL. On what the eigenvalues should be found in the parallel analysis. Can be one of "SMC", "PCA", or "EFA". If using "SMC" (default), the diagonal of the correlation matrices is replaced by the squared multiple correlations (SMCs) of the indicators. If using "PCA", the diagonal values of the correlation matrices are left to be 1. If using "EFA", eigenvalues are found on the correlation matrices with the final communalities of an EFA solution as diagonal.

eigen_type_KGC_PA
: character. Passed to KGC and PARALLEL. The same as eigen_type_HULL, but multiple inputs are possible here. Default is to use all inputs, that is, c("PCA","SMC","EFA")

n_factors
: numeric. Passed to PARALLEL (also within HULL) and to KGC. Number of factors to extract if "EFA" is included in eigen_type_HULL or eigen_type_KGC_PA. Default is 1.

n_datasets
: numeric. Passed to PARALLEL (also within HULL). The number of datasets to simulate. Default is 1000.

percent
: numeric. Passed to PARALLEL (also within HULL). A vector of percentiles to take the simulated eigenvalues from. Default is 95.

decision_rule
: character. Passed to PARALLEL (also within HULL). Which rule to use to determine the number of factors to retain. Default is "means", which will use the average simulated eigenvalues. "percentile", uses the percentiles specified in percent. "crawford" uses the 95th percentile for the first factor and the mean afterwards (based on Crawford et al, 2010).

...
: Further arguments passed to EFA in PARALLEL (also within HULL) and KGC.

Details

By default, the entered data are checked for suitability for factor analysis using the following methods (see respective documentations for details):

- Bartlett's test of sphericity (see BARTLETT)
- Kaiser-Meyer-Olkin criterion (see KMO)

The available factor retention criteria are the following (see respective documentations for details):

- Comparison data (see CD)
- Empirical Kaiser criterion (see EKC)
- Hull method (see HULL)
- Kaiser-Guttman criterion (see KGC)
- Parallel analysis (see PARALLEL)
- Sequential chi-square model tests, RMSEA lower bound, and AIC (see SMT)

Value

A list of class N_FACTORS containing

`outputs`	A list with the outputs from BARTLETT and KMO and the factor retention criteria.
`n_factors`	A named vector containing the suggested number of factors from each factor retention criterion.
`settings`	A list of the settings used.

Examples

```
# All criteria, with correlation matrix and fit method "ML" (where needed)
# This will throw a warning for CD, as no raw data were specified
nfac_all <- N_FACTORS(test_models$baseline$cormat, N = 500, method = "ML")

# The same as above, but without "CD"
nfac_wo_CD <- N_FACTORS(test_models$baseline$cormat, criteria = c("EKC",
                        "HULL", "KGC", "PARALLEL", "SMT"), N = 500,
                        method = "ML")

# Use PAF instead of ML (this will take a lot longer). For this, gof has
# to be set to "CAF" for the Hull method.
nfac_PAF <- N_FACTORS(test_models$baseline$cormat, criteria = c("EKC",
                      "HULL", "KGC", "PARALLEL", "SMT"), N = 500,
                      gof = "CAF")

# Do KGC and PARALLEL with only "PCA" type of eigenvalues
nfac_PCA <- N_FACTORS(test_models$baseline$cormat, criteria = c("EKC",
                      "HULL", "KGC", "PARALLEL", "SMT"), N = 500,
                      method = "ML", eigen_type_KGC_PA = "PCA")

# Use raw data, such that CD can also be performed
nfac_raw <- N_FACTORS(GRiPS_raw, method = "ML")
```

`OMEGA` *McDonald's omega*

Description

This function finds omega total, omega hierarchical, and omega subscale from a Schmid-Leiman (SL) solution or lavaan single factor or bifactor solution. The SL-based omegas can either be found from a `psych::schmid`, `SL`, or, in a more flexible way, by leaving `model = NULL` and specifying additional arguments. By setting the `type` argument, results from `psych::omega` can be reproduced.

Usage

```
OMEGA(
  model = NULL,
  type = c("EFAtools", "psych"),
  g_name = "g",
  group_names = NULL,
  factor_corres = NULL,
  var_names = NULL,
  fac_names = NULL,
  g_load = NULL,
  s_load = NULL,
  u2 = NULL,
  cormat = NULL,
  pattern = NULL,
  Phi = NULL,
  variance = c("correlation", "sums_load")
)
```

Arguments

model	class SL, class schmid, or class lavaan object. That is, an output object from SL or psych::schmid, or a lavaan fit object with a single factor or bifactor solution. If of class lavaan, only g_name needs to be specified additionally. If of class SL or schmid, only the arguments factor_corres and cormat need to be specified additionally.
type	character. Either "EFAtools" (default) or "psych" (see details)
g_name	character. The name of the general factor from the lavaan bifactor solution. This needs only be specified if model is a lavaan bifactor solution. Default is "g".
group_names	character. An optional vector of group names. The length must correspond to the number of groups for which the lavaan model was fitted.
factor_corres	numeric. A vector that indicates which variable corresponds to which group factor. Must be in the same order as the SL solution. For example c(3, 3, 3, 1, 1, 2, 2) if the first three variables load on the third group factor of the SL solution, the next two on the first group factor and the last two on the second group factor. If a variable should not be assigned to any group factor, insert a zero at its position (e.g. c(3, 3, 0, 1, 1, 2, 2), the third variable has no corresponding group factor).
var_names	character. A vector with subtest names in the order of the rows from the SL solution. This needs only be specified if model is left NULL.
fac_names	character. An optional vector of group factor names in the order of the columns of the SL solution. If left NULL, names of the group factors from the entered solution are taken.
g_load	numeric. A vector of general factor loadings from an SL solution. This needs only be specified if model is left NULL.
s_load	matrix. A matrix of group factor loadings from an SL solution. This needs only be specified if model is left NULL.

u2
: numeric. A vector of uniquenesses from an SL solution. This needs only be specified if model is left NULL.

cormat
: matrix. A correlation matrix to be used when variance = "correlation". If left NULL and an SL output is entered in model, the correlation matrix is taken from the output. If left NULL and a psych::schmid output is entered, the correlation matrix will be found based on the pattern matrix and Phi from the psych::schmid output using psych::factor.model. If left NULL and model is also left NULL, the correlation matrix is found based on the pattern matrix and Phi entered. However, if the correlation matrix is available, cormat should be specified instead of Phi and pattern.

pattern
: matrix. Pattern coefficients from an oblique factor solution. This needs only be specified if model is left NULL, variance = "correlation" and cormat is also left NULL.

Phi
: matrix. Factor intercorrelations from an oblique factor solution. This needs only be specified if model is left NULL, variance = "correlation" and cormat is also left NULL.

variance
: character. If "correlation" (default), then total variances for the whole scale as well as for the subscale composites are calculated based on the correlation matrix. If "sums_load", then total variances are calculated using the squared sums of general factor loadings and group factor loadings and the sum of uniquenesses (see details).

Details

If model is a lavaan bifactor solution, only the name of the general factor from the lavaan model needs to be specified additionally with the g_name argument. There is also the possibility to enter a lavaan single factor solution In this case, g_name is not needed. Finally, if a solution (bifactor or single factor) from a lavaan multiple group analysis is entered, the omegas are computed for each group. The type argument is not evaluated if model is of class lavaan.

If model is of class SL or psych::schmid only the type and, depending on the type (see below), the factor_corres arguments need to be specified additionally. If model is of class psych::schmid and variance = "correlation" (default), it is recommended to also provide the original correlation matrix in cormat to get more accurate results. Otherwise, the correlation matrix will be found based on the pattern matrix and Phi from the psych::schmid output using the psych::factor.model function.

If model = NULL, the arguments type, factor_corres (depending on the type, see below), var_names, g_load, s_load, and u2 and either cormat (recommended) or Phi and pattern need to be specified. If Phi and pattern are specified instead of cormat, the correlation matrix is found using the psych::factor.model function.

The only difference between type = "EFAtools" and type = "psych" is the determination of variable-to-factor correspondences. type = "psych" reproduces the psych::omega results, where variable-to-factor correspondences are found by taking the highest group factor loading for each variable as the relevant group factor loading. To do this, factor_corres must be left NULL.

The calculation of the total variance (for the whole scale as well as the subscale composites) can also be controlled in this function using the variance argument. For both types—"EFAtools" and "psych" —variance is set to "correlation" by default, which means that total variances are found using the correlation matrix. If `variance = "sums_load"` the total variance is calculated using the squared sums of general loadings and group factor loadings and the sum of the uniquenesses. This will only get comparable results to `variance = "correlation"` if no cross-loadings are present and simple structure is well-achieved in general with the SL solution (i.e., the uniquenesses should capture almost all of the variance not explained by the general factor and the variable's allocated group factor).

Value

If found for an SL or `lavaan` bifactor solution for one group: A matrix with omegas for the whole scale and for the subscales.

`tot`	Omega total.
`hier`	Omega hierarchical.
`sub`	Omega subscale.

If found for a `lavaan` single factor solution for one group: A vector with omega total for the single factor.

If found for a `lavaan` output from a multiple group analysis: A list containing the output described above for each group.

Source

McDonald, R. P. (1978). Generalizability in factorable domains: "Domain validity and generalizability". Educational and Psychological Measurement, 38, 75–79.

McDonald, R. P. (1985). Factor analysis and related methods. Hillsdale, NJ: Erlbaum.

McDonald, R. P. (1999). Test theory: A unified treatment. Mahwah, NJ: Erlbaum.

Gignac, G. E. (2014). On the Inappropriateness of Using Items to Calculate Total Scale Score Reliability via Coefficient Alpha for Multidimensional Scales. European Journal of Psychological Assessment, 30, 130-139.

Examples

```
## Use with a lavaan output

# Create and fit model in lavaan (assume all variables have SDs of 1)
mod <- 'F1 =~ V1 + V2 + V3 + V4 + V5 + V6
        F2 =~ V7 + V8 + V9 + V10 + V11 + V12
        F3 =~ V13 + V14 + V15 + V16 + V17 + V18
        g =~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 + V12 +
             V13 + V14 + V15 + V16 + V17 + V18'
fit <- lavaan::cfa(mod, sample.cov = test_models$baseline$cormat,
                   sample.nobs = 500, estimator = "ml", orthogonal = TRUE)

# Compute omega
OMEGA(fit, g_name = "g")
```

```
## Use with an output from the SL function, with type EFAtools
efa_mod <- EFA(test_models$baseline$cormat, N = 500, n_factors = 3,
               type = "EFAtools", method = "PAF", rotation = "promax")
sl_mod <- SL(efa_mod, type = "EFAtools", method = "PAF")

OMEGA(sl_mod, type = "EFAtools", factor_corres = rep(c(3, 2, 1), each = 6))

## Use with an output from the psych::schmid function, with type psych for
## OMEGA
schmid_mod <- psych::schmid(test_models$baseline$cormat, nfactors = 3,
                            n.obs = 500, fm = "pa", rotate = "Promax")
# Find correlation matrix from phi and pattern matrix from psych::schmid output
OMEGA(schmid_mod, type = "psych")
# Use specified correlation matrix
OMEGA(schmid_mod, type = "psych", cormat = test_models$baseline$cormat)

## Manually specify components (useful if omegas should be computed for a SL
## or bifactor solution found with another program)
## As an example, we extract the elements from an SL output here. This gives
## the same results as in the second example above.

efa_mod <- EFA(test_models$baseline$cormat, N = 500, n_factors = 3,
               type = "EFAtools", method = "PAF", rotation = "promax")
sl_mod <- SL(efa_mod, type = "EFAtools", method = "PAF")

OMEGA(model = NULL, type = "EFAtools", var_names = rownames(sl_mod$sl),
      g_load = sl_mod$sl[, "g"], s_load = sl_mod$sl[, c("F1", "F2", "F3")],
      u2 = sl_mod$sl[, "u2"], cormat = test_models$baseline$cormat,
      factor_corres = rep(c(3, 2, 1), each = 6))
```

PARALLEL *Parallel analysis*

Description

Various methods for performing parallel analysis. This function uses future_lapply for which a parallel processing plan can be selected. To do so, call `library(future)` and, for example, `plan(multisession)`; see examples.

Usage

```
PARALLEL(
  x = NULL,
  N = NA,
  n_vars = NA,
  n_datasets = 1000,
  percent = 95,
  eigen_type = c("PCA", "SMC", "EFA"),
```

```
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall"),
  decision_rule = c("means", "percentile", "crawford"),
  n_factors = 1,
  ...
)
```

Arguments

x	matrix or data.frame. The real data to compare the simulated eigenvalues against. Must not contain variables of classes other than numeric. Can be a correlation matrix or raw data.
N	numeric. The number of cases / observations to simulate. Only has to be specified if `x` is either a correlation matrix or `NULL`. If x contains raw data, `N` is found from the dimensions of `x`.
n_vars	numeric. The number of variables / indicators to simulate. Only has to be specified if `x` is left as `NULL` as otherwise the dimensions are taken from `x`.
n_datasets	numeric. The number of datasets to simulate. Default is 1000.
percent	numeric. A vector of percentiles to take the simulated eigenvalues from. Default is 95.
eigen_type	character. On what the eigenvalues should be found. Can be either "SMC", "PCA", or "EFA". If using "SMC", the diagonal of the correlation matrix is replaced by the squared multiple correlations (SMCs) of the indicators. If using "PCA", the diagonal values of the correlation matrices are left to be 1. If using "EFA", eigenvalues are found on the correlation matrices with the final communalities of an EFA solution as diagonal.
use	character. Passed to `stats::cor` if raw data is given as input. Default is "pairwise.complete.obs".
cor_method	character. Passed to `stats::cor` Default is "pearson".
decision_rule	character. Which rule to use to determine the number of factors to retain. Default is `"means"`, which will use the average simulated eigenvalues. `"percentile"`, uses the percentiles specified in percent. `"crawford"` uses the 95th percentile for the first factor and the mean afterwards (based on Crawford et al, 2010).
n_factors	numeric. Number of factors to extract if "EFA" is included in `eigen_type`. Default is 1.
...	Additional arguments passed to `EFA`. For example, the extraction method can be changed here (default is "PAF"). PAF is more robust, but it will take longer compared to the other estimation methods available ("ML" and "ULS").

Details

Parallel analysis (Horn, 1965) compares the eigenvalues obtained from the sample correlation matrix against those of null model correlation matrices (i.e., with uncorrelated variables) of the same sample size. This way, it accounts for the variation in eigenvalues introduced by sampling error and thus eliminates the main problem inherent in the Kaiser-Guttman criterion (`KGC`).

Three different ways of finding the eigenvalues under the factor model are implemented, namely "SMC", "PCA", and "EFA". PCA leaves the diagonal elements of the correlation matrix as they are and is thus equivalent to what is done in PCA. SMC uses squared multiple correlations as communality estimates with which the diagonal of the correlation matrix is replaced. Finally, EFA performs an `EFA` with one factor (can be adapted to more factors) to estimate the communalities and based on the correlation matrix with these as diagonal elements, finds the eigenvalues.

Parallel analysis is often argued to be one of the most accurate factor retention criteria. However, for highly correlated factor structures it has been shown to underestimate the correct number of factors. The reason for this is that a null model (uncorrelated variables) is used as reference. However, when factors are highly correlated, the first eigenvalue will be much larger compared to the following ones, as later eigenvalues are conditional on the earlier ones in the sequence and thus the shared variance is already accounted in the first eigenvalue (e.g., Braeken & van Assen, 2017).

The `PARALLEL` function can also be called together with other factor retention criteria in the `N_FACTORS` function.

Value

A list of class PARALLEL containing the following objects

`eigenvalues_PCA`
: A matrix containing the eigenvalues of the real and the simulated data found with eigen_type = "PCA"

`eigenvalues_SMC`
: A matrix containing the eigenvalues of the real and the simulated data found with eigen_type = "SMC"

`eigenvalues_EFA`
: A matrix containing the eigenvalues of the real and the simulated data found with eigen_type = "EFA"

`n_fac_PCA`
: The number of factors to retain according to the parallel procedure with eigen_type = "PCA".

`n_fac_SMC`
: The number of factors to retain according to the parallel procedure with eigen_type = "SMC".

`n_fac_EFA`
: The number of factors to retain according to the parallel procedure with eigen_type = "EFA".

`settings`
: A list of control settings used in the print function.

Source

Braeken, J., & van Assen, M. A. (2017). An empirical Kaiser criterion. Psychological Methods, 22, 450 – 466. http://dx.doi.org/10.1037/ met0000074

Crawford, A. V., Green, S. B., Levy, R., Lo, W. J., Scott, L., Svetina, D., & Thompson, M. S. (2010). Evaluation of parallel analysis methods for determining the number of factors. Educational and Psychological Measurement, 70(6), 885-901.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. Psychometrika, 30(2), 179–185. doi: 10.1007/BF02289447

See Also

Other factor retention criteria: CD, EKC, HULL, KGC, SMT

N_FACTORS as a wrapper function for this and all the above-mentioned factor retention criteria.

Examples

```
# example without real data
pa_unreal <- PARALLEL(N = 500, n_vars = 10)

# example with correlation matrix with all eigen_types and PAF estimation
pa_paf <- PARALLEL(test_models$case_11b$cormat, N = 500)

# example with correlation matrix with all eigen_types and ML estimation
# this will be faster than the above with PAF)
pa_ml <- PARALLEL(test_models$case_11b$cormat, N = 500, method = "ML")


## Not run:
# for parallel computation
future::plan(future::multisession)
pa_faster <- PARALLEL(test_models$case_11b$cormat, N = 500)

## End(Not run)
```

---

plot.CD *Plot CD object*

Description

Plot method showing a summarized output of the CD function

Usage

```
## S3 method for class 'CD'
plot(x, ...)
```

Arguments

x	a list of class CD. An output from the CD function.
...	not used.

plot.EKC *Plot EKC object*

Description

Plot method showing a summarized output of the EKC function

Usage

```
## S3 method for class 'EKC'
plot(x, ...)
```

Arguments

x	a list of class EKC. An output from the EKC function.
...	not used.

Examples

```
EKC_base <- EKC(test_models$baseline$cormat, N = 500)
plot(EKC_base)
```

plot.HULL *Plot HULL object*

Description

Plot method showing a summarized output of the `HULL` function

Usage

```
## S3 method for class 'HULL'
plot(x, ...)
```

Arguments

x	list of class HULL. An output from the `HULL` function.
...	not used.

Examples

```
x <- HULL(test_models$baseline$cormat, N = 500, method = "ML")
plot(x)
```

plot.KGC *Plot KGC object*

Description

Plot method showing a summarized output of the KGC function

Usage

```
## S3 method for class 'KGC'
plot(x, ...)
```

Arguments

x	a list of class KGC. An output from the KGC function.
...	not used.

Examples

```
KGC_base <- KGC(test_models$baseline$cormat)
plot(KGC_base)
```

plot.PARALLEL *Plot PARALLEL object*

Description

Plot method showing a summarized output of the PARALLEL function

Usage

```
## S3 method for class 'PARALLEL'
plot(x, ...)
```

Arguments

x	list of class PARALLEL. An output from the PARALLEL function.
...	not used.

Examples

```
# example with correlation matrix and "ML" estimation
x <- PARALLEL(test_models$case_11b$cormat, N = 500, method = "ML")
plot(x)
```

population_models *population_models*

Description

Population factor models, some of which (baseline to case_11e) used for the simulation analyses reported in Grieder and Steiner (2019). All combinations of the pattern matrices and the factor intercorrelations were used in the simulations. Many models are based on cases used in de Winter and Dodou (2012).

Usage

```
population_models
```

Format

A list of 3 lists "loadings", "phis_3", and "phis_6".

`loadings` contains the following matrices of pattern coefficients:

baseline (matrix) - The pattern coefficients of the baseline model. Three factors with six indicators each, all with pattern coefficients of .6. Same baseline model as used in de Winter and Dodou (2012).

case_1a (matrix) - Three factors with 2 indicators per factor.

case_1b (matrix) - Three factors with 3 indicators per factor. Case 5 in de Winter and Dodou (2012).

case_1c (matrix) - Three factors with 4 indicators per factor.

case_1d (matrix) - Three factors with 5 indicators per factor.

case_2 (matrix) - Same as baseline model but with low pattern coefficients of .3.

case_3 (matrix) - Same as baseline model but with high pattern coefficients of .9.

case_4 (matrix) - Three factors with different pattern coefficients *between* factors (one factor with .9, one with .6, and one with .3, respectively). Case 7 in de Winter and Dodou (2012).

case_5 (matrix) - Three factors with different pattern coefficients *within* factors (each factor has two pattern coefficients of each .9, .6, and .3). Similar to cases 8/ 9 in de Winter and Dodou (2012).

case_6a (matrix) - Same as baseline model but with one cross loading of .4. Similar to case 10 in de Winter and Dodou (2012).

case_6b (matrix) - Same as baseline model but with three cross loading of .4 (One factor with 2 and one with 1 crossloading). Similar to case 10 in de Winter and Dodou (2012).

case_7 (matrix) - Three factors with different number of indicators per factor (2, 4, and 6 respectively). Similar to cases 11/ 12 in de Winter and Dodou (2012).

case_8 (matrix) - Three factors with random variation in pattern coefficients added, drawn from a uniform distribution between [-.2, .2]. Case 13 in de Winter and Dodou (2012).

case_9a (matrix) - Three factors with 2 indicators per factor, with different pattern coefficients within one of the factors.

case_9b (matrix) - Three factors with 3 indicators per factor, with different pattern coefficients.

case_9c (matrix) - Three factors with 4 indicators per factor, with different pattern coefficients.

case_9d (matrix) - Three factors with 5 indicators per factor, with different pattern coefficients.

case_10a (matrix) - Six factors with 2 indicators per factor, all with pattern coefficients of .6.

case_10b (matrix) - Six factors with 3 indicators per factor, all with pattern coefficients of .6.

case_10c (matrix) - Six factors with 4 indicators per factor, all with pattern coefficients of .6.

case_10d (matrix) - Six factors with 5 indicators per factor, all with pattern coefficients of .6.

case_10e (matrix) - Six factors with 6 indicators per factor, all with pattern coefficients of .6.

case_11a (matrix) - Six factors with 2 indicators per factor, with different pattern coefficients within and between factors (.3, .6, and .9).

case_11b (matrix) - Six factors with 3 indicators per factor, with different pattern coefficients within and between factors (.3, .6, and .9).

case_11c (matrix) - Six factors with 4 indicators per factor, with different pattern coefficients within and between factors (.3, .6, and .9).

case_11d (matrix) - Six factors with 5 indicators per factor, with different pattern coefficients within and between factors (.3, .6, and .9).

case_11e (matrix) - Six factors with 6 indicators per factor, with different pattern coefficients within and between factors (.3, .6, and .9).

case_12a (matrix) - One factor, with 2 equal pattern coefficients (.6).

case_12b (matrix) - One factor, with 3 equal pattern coefficients (.6).

case_12c (matrix) - One factor, with 6 equal pattern coefficients (.6).

case_12d (matrix) - One factor, with 10 equal pattern coefficients (.6).

case_12e (matrix) - One factor, with 15 equal pattern coefficients (.6).

case_13a (matrix) - One factor, with 2 different pattern coefficients (.3, and .6).

case_13b (matrix) - One factor, with 3 different pattern coefficients (.3, .6, and .9).

case_13c (matrix) - One factor, with 6 different pattern coefficients (.3, .6, and .9).

case_13d (matrix) - One factor, with 10 different pattern coefficients (.3, .6, and .9).

case_13e (matrix) - One factor, with 15 different pattern coefficients (.3, .6, and .9).

case_14a (matrix) - No factor, 2 variables (0).

case_14b (matrix) - No factor, 3 variables (0).

case_14c (matrix) - No factor, 6 variables (0).

case_14d (matrix) - No factor, 10 variables (0).

case_14e (matrix) - No factor, 15 variables (0). `phis_3` contains the following 3x3 matrices:

zero (matrix) - Matrix of factor intercorrelations of 0. Same intercorrelations as used in de Winter and Dodou (2012).

moderate (matrix) - Matrix of moderate factor intercorrelations of .3.

mixed (matrix) - Matrix of mixed (.3, .5, and .7) factor intercorrelations.

strong (matrix) - Matrix of strong factor intercorrelations of .7. Same intercorrelations as used in de Winter and Dodou (2012). `phis_6` contains the following 6x6 matrices:

zero (matrix) - Matrix of factor intercorrelations of 0. Same intercorrelations as used in de Winter and Dodou (2012).

moderate (matrix) - Matrix of moderate factor intercorrelations of .3.

mixed (matrix) - Matrix of mixed (around .3, .5, and .7; smoothing was necessary for the matrix to be positive definite) factor intercorrelations.

strong (matrix) - Matrix of strong factor intercorrelations of .7. Same intercorrelations as used in de Winter and Dodou (2012).

Source

Grieder, S., & Steiner, M.D. (2020). Algorithmic Jingle Jungle: A Comparison of Implementations of Principal Axis Factoring and Promax Rotation in R and SPSS. Manuscript in Preparation.

de Winter, J.C.F., & Dodou, D. (2012). Factor recovery by principal axis factoring and maximum likelihood factor analysis as a function of factor pattern and sample size. Journal of Applied Statistics. 39.

`print.BARTLETT` *Print BARTLETT object*

Description

Print BARTLETT object

Usage

```
## S3 method for class 'BARTLETT'
print(x, ...)
```

Arguments

`x`	list of class BARTLETT (output from the BARTLETT function)
`...`	additional arguments passed to print

Examples

```
BARTLETT(test_models$baseline$cormat, N = 500)
```

print.CD *Print function for CD objects*

Description

Print function for CD objects

Usage

```
## S3 method for class 'CD'
print(x, plot = TRUE, ...)
```

Arguments

x	a list of class CD. Output from CD function.
plot	logical. Whether to plot the results.
...	Further arguments for print.

Examples

```
# determine n factors of the GRiPS
CD(GRiPS_raw)
```

print.COMPARE *Print COMPARE object*

Description

Print Method showing a summarized output of the `COMPARE` function.

Usage

```
## S3 method for class 'COMPARE'
print(x, ...)
```

Arguments

x	list. An object of class COMPARE to be printed
...	Further arguments for print.

Examples

```
# A type SPSS EFA to mimick the SPSS implementation
EFA_SPSS_5 <- EFA(IDS2_R, n_factors = 5, type = "SPSS")

# A type psych EFA to mimick the psych::fa() implementation
EFA_psych_5 <- EFA(IDS2_R, n_factors = 5, type = "psych")

# compare the two
COMPARE(EFA_SPSS_5$unrot_loadings, EFA_psych_5$unrot_loadings,
        x_labels = c("SPSS", "psych"))
```

---

print.EFA *Print EFA object*

Description

Print Method showing a summarized output of the EFA function

Usage

```
## S3 method for class 'EFA'
print(x, ...)
```

Arguments

x	list. An object of class EFA to be printed
...	Further arguments for print.

Examples

```
EFAtools_PAF <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                    type = "EFAtools", method = "PAF", rotation = "promax")
EFAtools_PAF
```

---

print.EKC *Print function for EKC objects*

Description

Print function for EKC objects

Usage

```
## S3 method for class 'EKC'
print(x, plot = TRUE, ...)
```

Arguments

`x`	a list of class EKC. Output from `EKC` function.
`plot`	logical. Whether to plot the results.
`...`	Further arguments for print.

Examples

```
EKC_base <- EKC(test_models$baseline$cormat, N = 500)
EKC_base
```

`print.HULL` *Print function for HULL objects*

Description

Print function for HULL objects

Usage

```
## S3 method for class 'HULL'
print(x, plot = TRUE, ...)
```

Arguments

`x`	a list of class HULL. Output from the `HULL` function.
`plot`	logical. Whether to plot the results.
`...`	Further arguments for print.

Examples

```
HULL(test_models$baseline$cormat, N = 500, method = "ML")
```

print.KGC *Print function for KGC objects*

Description

Print function for KGC objects

Usage

```
## S3 method for class 'KGC'
print(x, plot = TRUE, ...)
```

Arguments

x	a list of class KGC. Output from KGC function.
plot	logical. Whether to plot the results.
...	Further arguments for print.

Examples

```
KGC_base <- KGC(test_models$baseline$cormat)
KGC_base
```

print.KMO *Print KMO object*

Description

Print KMO object

Usage

```
## S3 method for class 'KMO'
print(x, ...)
```

Arguments

x	list of class KMO (output from the KMO function)
...	additional arguments passed to print

Examples

```
KMO_base <- KMO(test_models$baseline$cormat)
KMO_base
```

print.LOADINGS *Print LOADINGS object*

Description

Print LOADINGS object

Usage

```
## S3 method for class 'LOADINGS'
print(x, cutoff = 0.3, digits = 3, ...)
```

Arguments

x	class LOADINGS matrix.
cutoff	numeric. The number above which to print loadings in bold default is .3.
digits	numeric. Passed to round. Number of digits to round the loadings to (default is 3).
...	additional arguments passed to print

Examples

```
EFAtools_PAF <- EFA(test_models$baseline$cormat, n_factors = 3, N = 500,
                    type = "EFAtools", method = "PAF", rotation = "promax")
EFAtools_PAF
```

print.N_FACTORS *Print function for N_FACTORS objects*

Description

Print function for N_FACTORS objects

Usage

```
## S3 method for class 'N_FACTORS'
print(x, ...)
```

Arguments

x	a list of class N_FACTORS. Output from N_FACTORS function.
...	Further arguments for print.

Examples

```
# All criteria except "CD", with correlation matrix and fit method "ML"
# (where needed)
N_FACTORS(test_models$baseline$cormat, criteria = c("EKC", "HULL", "KGC",
          "PARALLEL", "SMT"), N = 500, method = "ML")
```

---

print.OMEGA *Print OMEGA object*

Description

Print OMEGA object

Usage

```
## S3 method for class 'OMEGA'
print(x, digits = 3, ...)
```

Arguments

x	output of class OMEGA (output from the OMEGA function)
digits	numeric. Passed to `round`. Number of digits to round to (default is 3).
...	additional arguments passed to print

Examples

```
efa_mod <- EFA(test_models$baseline$cormat, N = 500, n_factors = 3,
               type = "EFAtools", method = "PAF", rotation = "promax")
sl_mod <- SL(efa_mod, type = "EFAtools", method = "PAF")

OMEGA(sl_mod, type = "EFAtools", factor_corres = rep(c(3, 2, 1), each = 6))
```

---

print.PARALLEL *Print function for PARALLEL objects*

Description

Print function for PARALLEL objects

Usage

```
## S3 method for class 'PARALLEL'
print(x, plot = TRUE, ...)
```

Arguments

x	a list of class PARALLEL. Output from PARALLEL function.
plot	logical. Whether to plot the results.
...	Further arguments for print.

Examples

```
# example without real data
PARALLEL(N = 500, n_vars = 10)

# example with correlation matrix and "ML" estimation
PARALLEL(test_models$case_11b$cormat, N = 500, method = "ML")
```

---

print.SL *Print SL object*

Description

Print Method showing a summarized output of the SL function.

Usage

```
## S3 method for class 'SL'
print(x, ...)
```

Arguments

x	list. An object of class SL to be printed
...	Further arguments for print.

Examples

```
EFA_mod <- EFA(test_models$baseline$cormat, N = 500, n_factors = 3,
               type = "EFAtools", method = "PAF", rotation = "promax")
SL(EFA_mod, type = "EFAtools", method = "PAF")
```

print.SLLOADINGS *Print SLLOADINGS object*

Description

Print SLLOADINGS object

Usage

```
## S3 method for class 'SLLOADINGS'
print(x, cutoff = 0.2, digits = 3, ...)
```

Arguments

`x`	class SLLOADINGS matrix.
`cutoff`	numeric. The number above which to print loadings in bold (default is .2).
`digits`	numeric. Passed to round. Number of digits to round the loadings to (default is 3).
`...`	additional arguments passed to print

Examples

```
EFA_mod <- EFA(test_models$baseline$cormat, N = 500, n_factors = 3,
               type = "EFAtools", method = "PAF", rotation = "promax")
SL(EFA_mod, type = "EFAtools", method = "PAF")
```

print.SMT *Print SMT object*

Description

Print SMT object

Usage

```
## S3 method for class 'SMT'
print(x, ...)
```

Arguments

`x`	list of class SMT (output from the SMT function)
`...`	additional arguments passed to print

Examples

```
SMT_base <- SMT(test_models$baseline$cormat, N = 500)
SMT_base
```

RiskDimensions *RiskDimensions*

Description

A list containing the bivariate correlations (cormat) of the 9 dimensions on which participants in Fischhoff et al. (1978) rated different activities and technologies as well as the sample size (N). This was then analyzed together with ratings of the risks and benefits of these activities and technologies.

Usage

```
RiskDimensions
```

Format

An object of class `list` of length 2.

Source

Fischhoff, B, Slovic, P, Lichtenstein, S, Read, S, and Combs, B. (1978). How safe is safe enough? A psychometric study of attitudes towards technological risks and benefits. Policy Sciences, 9, 127-152. doi: 10.1007/BF00143739

SL *Schmid-Leiman Transformation*

Description

This function implements the Schmid-Leiman (SL) transformation (Schmid & Leiman, 1957). It takes the pattern coefficients and factor intercorrelations from an oblique factor solution as input and can reproduce the results from `psych::schmid` and from the SPSS implementation from Wolff & Preising (2005). Other arguments from `EFA` can be used to control the procedure to find the second-order loadings more flexibly.

Usage

```
SL(
  x,
  Phi = NULL,
  type = c("EFAtools", "psych", "SPSS", "none"),
  method = c("PAF", "ML", "ULS"),
  ...
)
```

Arguments

x	object of class EFA or class psych::fa or matrix. If class EFA or class psych::fa, pattern coefficients and factor intercorrelations are taken from this object. x can also be a pattern matrix from an oblique factor solution (see Phi).
Phi	matrix. A matrix of factor intercorrelations from an oblique factor solution. Only needs to be specified if a pattern matrix is entered directly into x.
type	character. One of "EFAtools" (default), "psych", "SPSS", or "none". This is used to control the procedure of the second-order factor analysis. See EFA for details.
method	character. One of "PAF", "ML", or "ULS" to use principal axis factoring, maximum likelihood, or unweighted least squares (also called minres), respectively, used in EFA to find the second-order loadings.
...	Arguments to be passed to EFA.

Details

The SL transformation (also called SL orthogonalization) is a procedure with which an oblique factor solution is transformed into a hierarchical, orthogonalized solution. As a first step, the factor intercorrelations are again factor analyzed to find second-order factor loadings. If there is only one higher-order factor, this step of the procedure stops there, resulting in a second-order factor structure. The first-order factor and the second-order factor are then orthogonalized, resulting in an orthogonalized factor solution with proportionality constraints. The procedure thus makes a suggested hierarchical data structure based on factor intercorrelations explicit. One major advantage of SL transformation is that it enables variance partitioning between higher-order and first-order factors, including the calculation of McDonald's omegas (see OMEGA).

Value

A list of class SL containing the following

orig_R	Original correlation matrix.
sl	A matrix with general factor loadings, group factor loadings, communalities, and uniquenesses.
L2	Second-order factor loadings.
vars_accounted	A matrix of explained variances and sums of squared loadings.
iter	The number of iterations needed for convergence in EFA.
settings	list. The settings (arguments) used in EFA to get the second-order loadings.

Source

Schmid, J. & Leiman, J. M. (1957). The development of hierarchical factor solutions. Psychometrika, 22(1), 53–61. doi:10.1007/BF02289209

Wolff, H.-G., & Preising, K. (2005). Exploring item and higher order factor structure with the Schmid-Leiman solution: Syntax codes for SPSS and SAS. Behavior Research Methods, 37 , 48–58. doi:10.3758/BF03206397

Examples

```
## Use with an output from the EFAtools::EFA function, both with type EFAtools
EFA_mod <- EFA(test_models$baseline$cormat, N = 500, n_factors = 3,
               type = "EFAtools", method = "PAF", rotation = "promax")
SL_EFAtools <- SL(EFA_mod, type = "EFAtools", method = "PAF")


## Use with an output from the psych::fa function with type psych in SL
fa_mod <- psych::fa(test_models$baseline$cormat, nfactors = 3, n.obs = 500,
                    fm = "pa", rotate = "Promax")
SL_psych <- SL(fa_mod, type = "psych", method = "PAF")


## Use more flexibly by entering a pattern matrix and phi directly (useful if
## a factor solution found with another program should be subjected to SL
## transformation)

## For demonstration, take pattern matrix and phi from an EFA output
## This gives the same solution as the first example
EFA_mod <- EFA(test_models$baseline$cormat, N = 500, n_factors = 3,
               type = "EFAtools", method = "PAF", rotation = "promax")
SL_flex <- SL(EFA_mod$rot_loadings, Phi = EFA_mod$Phi, type = "EFAtools",
              method = "PAF")
```

SMT *Sequential Chi Square Model Tests, RMSEA lower bound, and AIC*

Description

Sequential Chi Square Model Tests (SMT) are a factor retention method where multiple EFAs with increasing numbers of factors are fitted and the number of factors for which the Chi Square value first becomes non-significant is taken as the suggested number of factors. Preacher, Zhang, Kim, & Mels (2013) suggested a similar approach with the lower bound of the 90% confidence interval of the Root Mean Square Error of Approximation (RMSEA; Browne & Cudeck, 1992; Steiger & Lind, 1980), and with the Akaike Information Criterion (AIC). For the RMSEA, the number of factors for which this lower bound first falls below .05 is the suggested number of factors to retain. For the AIC, it is the number of factors where the AIC is lowest.

Usage

```
SMT(
  x,
  N = NA,
  use = c("pairwise.complete.obs", "all.obs", "complete.obs", "everything",
    "na.or.complete"),
  cor_method = c("pearson", "spearman", "kendall")
)
```

Arguments

`x`	data.frame or matrix. Dataframe or matrix of raw data or matrix with correlations.
`N`	numeric. The number of observations. Needs only be specified if a correlation matrix is used.
`use`	character. Passed to `stats::cor` if raw data is given as input. Default is "pairwise.complete.obs".
`cor_method`	character. Passed to `stats::cor`. Default is "pearson".

Details

As a first step in the procedure, a maximum number of factors to extract is determined for which the model is still over-identified (df > 0).

Then, EFAs with increasing numbers of factors from 1 to the maximum number are fitted with maximum likelihood estimation.

For the SMT, first the significance of the chi square value for a model with 0 factors is determined. If this value is not significant, 0 factors are suggested to retain. If it is significant, a model with 1 factor is estimated and the significance of its chi square value is determined, and so on, until a non-significant result is obtained. The suggested number of factors is the number of factors for the model where the chi square value first becomes non-significant.

Regarding the RMSEA, the suggested number of factors is the number of factors for the model where the lower bound of the 90% confidence interval of the RMSEA first falls below the .05 threshold.

Regarding the AIC, the suggested number of factors is the number of factors for the model with the lowest AIC.

In comparison with other prominent factor retention criteria, SMT performed well at determining the number of factors to extract in EFA (Auerswald & Moshagen, 2019). The RMSEA lower bound also performed well at determining the true number of factors, while the AIC performed well at determining the most generalizable model (Preacher, Zhang, Kim, & Mels, 2013).

The `SMT` function can also be called together with other factor retention criteria in the `N_FACTORS` function.

Value

A list of class SMT containing

`nfac_chi`	The number of factors to retain according to the significance of the chi square value.
`nfac_RMSEA`	The number of factors to retain according to the RMSEA lower bound
`nfac_AIC`	The number of factors to retain according to the AIC
`p_null`	The p-value for the null model (zero factors)
`ps_chi`	The p-values for EFA models with increasing numbers of factors, starting with 1 factor
`RMSEA_LB_null`	The lower bounds of the 90% confidence interval for the RMSEA for the null model (zero factors).

`RMSEA_LBs`	The lower bounds of the 90% confidence interval for the RMSEA for EFA models with increasing numbers of factors, starting with 1 factor
`AIC_null`	The AICs for the null model (zero factors)
`AICs`	The AICs for EFA models with increasing numbers of factors, starting with 1 factor

Source

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. Psychological Methods, 24(4), 468–491. https://doi.org/10.1037/met0000200

Browne, M.W., & Cudeck, R. (1992). Alternative ways of assessing model fit. Sociological Methods and Research, 21, 230–258.

Preacher, K. J., Zhang G., Kim, C., & Mels, G. (2013). Choosing the Optimal Number of Factors in Exploratory Factor Analysis: A Model Selection Perspective, Multivariate Behavioral Research, 48(1), 28-56, doi:10.108/00273171.2012.710386

Steiger, J. H., & Lind, J. C. (1980, May). Statistically based tests for the number of common factors. Paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

See Also

Other factor retention criteria: `CD`, `EKC`, `HULL`, `KGC`, `PARALLEL`

`N_FACTORS` as a wrapper function for this and all the above-mentioned factor retention criteria.

Examples

```
SMT_base <- SMT(test_models$baseline$cormat, N = 500)
SMT_base
```

`SPSS` *Various outputs from SPSS FACTOR*

Description

Various outputs from SPSS FACTOR for the IDS-2 (Grob & Hagmann-von Arx, 2018), the WJIV (3 to 5 and 20 to 39 years; McGrew, LaForte, & Schrank, 2014), the DOSPERT (Frey et al., 2017; Weber, Blais, & Betz, 2002), the NEO-PI-R (Costa, & McCrae, 1992), and four simulated datasets (baseline, case_1a, case_6b, and case_11b, see test_models and population_models) used in Grieder and Steiner (2020).

Usage

```
SPSS
```

Format

A list of 9 containing EFA results for each of the data sets mentioned above. Each of these nine entries is a list of 4 or 8 (see details), of the following structure:

paf_comm (vector) - The final communalities obtained with the FACTOR algorithm with PAF and no rotation. For details, see Grieder and Grob (2019).

paf_load (matrix) - F1 to FN = unrotated factor loadings obtained with the FACTOR algorithm with PAF. Rownames are the abbreviated subtest names.

paf_iter (numeric) - Number of iterations needed for the principal axis factoring to converge.

var_load (matrix) - F1 to FN = varimax rotated factor loadings obtained with the FACTOR algorithm with PAF. Rownames are the abbreviated subtest names.

pro_load (matrix) - F1 to FN = promax rotated factor loadings obtained with the FACTOR algorithm with PAF. Rownames are the abbreviated subtest names.

pro_phi (matrix) - F1 to FN = intercorrelations of the promax rotated loadings.

sl (matrix) - g = General / second order factor of the Schmid-Leiman solution. F1 to FN = First order factors of the Schmid-Leiman solution. h2 = Communalities of the Schmid-Leiman solution. This Schmid-Leiman solution was found using the SPSS Syntax provided by Wolff and Preising (2005).

L2 (matrix) - Second order loadings used for the Schmid-Leiman transformation. This Schmid-Leiman solution was found using the SPSS Syntax provided by Wolff and Preising (2005).

Details

The IDS-2, the two WJIV, the DOSPERT, and the NEO-PI-R contain all the above entries, while the four simulated datasets contain only paf_load, var_load, pro_load, and pro_phi.

Source

Grieder, S., & Steiner, M.D. (2020). Algorithmic Jingle Jungle: A Comparison of Implementations of Principal Axis Factoring and Promax Rotation in R and SPSS. Manuscript in Preparation.

Wolff, H.G., & Preising, K. (2005). Exploring item and higher order factor structure with the Schmid-Leiman solution: Syntax codes for SPSS and SAS. Behavior Research Methods, 37, 48–58. doi: 10.3758/BF03206397

Grieder, S., & Grob, A. (2019). Exploratory factor analyses of the intelligence and development scales–2: Implications for theory and practice. Assessment. Advance online publication. doi:10.1177/10731911198450

Grob, A., & Hagmann-von Arx, P. (2018). Intelligence and Development Scales–2 (IDS-2). Intelligenz- und Entwicklungsskalen für Kinder und Jugendliche. [Intelligence and Development Scales for Children and Adolescents.]. Bern, Switzerland: Hogrefe.

Frey, R., Pedroni, A., Mata, R., Rieskamp, J., & Hertwig, R. (2017). Risk preference shares the psychometric structure of major psychological traits. Science Advances, 3, e1701381.

McGrew, K. S., LaForte, E. M., & Schrank, F. A. (2014). Technical Manual. Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Costa, P. T., & McCrae, R. R. (1992). NEO PI-R professional manual. Odessa, FL: Psychological Assessment Resources, Inc.

`test_models` *Four test models used in Grieder and Steiner (2020)*

Description

Correlation matrices created from simulated data from four of the `population_models` cases, each with strong factor intercorrelations. These are used in Grieder & Steiner (2020) to compare the psych and SPSS implementations in this package with the actual implementations of the programs. For details on the cases, see `population_models`.

Usage

```
test_models
```

Format

A list of 4 lists "baseline", "case_1a", "case_6b", and"case_11b", each with the following elements.

cormat (matrix) - The correlation matrix of the simulated data.

n_factors (numeric) - The true number of factors.

N (numeric) - The sample size of the generated data.

Source

Grieder, S., & Steiner, M.D. (2020). Algorithmic Jingle Jungle: A Comparison of Implementations of Principal Axis Factoring and Promax Rotation in R and SPSS. Manuscript in Preparation.

`UPPS_raw` *UPPS_raw*

Description

A dataframe containing responses to the UPPS personality scale (Whiteside & Lynam, 2005) of 645 participants of Study 2 of Steiner and Frey (2020). Each column are the ratings to one of 45 items to assess urgency, premeditation, perseverance, and sensation seeking. The original data can be accessed via `https://osf.io/kxp8t/`.

Usage

```
UPPS_raw
```

Format

An object of class `data.frame` with 645 rows and 45 columns.

Source

Whiteside, S. P., Lynam, D. R., Miller, J. D., & Reynolds, S. K. (2005). Validation of the UPPS impulsive behaviour scale: A four-factor model of impulsivity. European Journal of Personality, 19 (7), 559–574.

Steiner, M., & Frey, R. (2020). Representative design in psychological assessment: A case study using the Balloon Analogue Risk Task (BART). PsyArXiv Preprint. doi:10.31234/osf.io/dg4ks

---

`WJIV_ages_14_19` *Woodcock Johnson IV: ages 14 to 19*

Description

A list containing the bivariate correlations (N = 1,685) of the 47 intelligence subtests from the WJ IV for 14- to 19-year-olds obtained from the WJ-IV technical manual (McGrew, LaForte, & Schrank, 2014). Tables are reproduced with permission from the publisher.

Usage

```
WJIV_ages_14_19
```

Format

A list of 2 with elements "cormat" (47 x 47 matrix of bivariate correlations) and "N" (scalar). The correlation matrix contains the following variables:

ORLVOC (numeric) - Oral Vocabulary.

NUMSER (numeric) - Number Series.

VRBATN (numeric) - Verbal Attention.

LETPAT (numeric) - Letter-Pattern Matching.

PHNPRO (numeric) - Phonological Processing.

STYREC (numeric) - Story Recall.

VISUAL (numeric) - Visualization.

GENINF (numeric) - General Information.

CONFRM (numeric) - Concept Formation.

NUMREV (numeric) - Numbers Reversed.

NUMPAT (numeric) - Number-Pattern Matching.

NWDREP (numeric) - Nonword Repetition.

VAL (numeric) - Visual-Auditory Learning.

PICREC (numeric) - Picture Recognition.

ANLSYN (numeric) - Analysis-Synthesis.

OBJNUM (numeric) - Object-Number Sequencing.

PAIRCN (numeric) - Pair Cancellation.

MEMWRD (numeric) - Memory for Words.

PICVOC (numeric) - Picture Vocabulary.

ORLCMP (numeric) - Oral Comprehension.

SEGMNT (numeric) - Segmentation.

RPCNAM (numeric) - Rapid Picture Naming.

SENREP (numeric) - Sentence Repetition.

UNDDIR (numeric) - Understanding Directions.

SNDBLN (numeric) - Sound Blending.

RETFLU (numeric) - Retrieval Fluency.

SNDAWR (numeric) - Sound Awareness.

LWIDNT (numeric) - Letter-Word Identification.

APPROB (numeric) - Applied Problems.

SPELL (numeric) - Spelling.

PSGCMP (numeric) - Passage Comprehension.

CALC (numeric) - Calculation.

WRTSMP (numeric) - Writing Samples.

WRDATK (numeric) - Word Attack.

ORLRDG (numeric) - Oral Reading.

SNRDFL (numeric) - Sentence Reading Fluency.

MTHFLU (numeric) - Math Facts Fluency.

SNWRFL (numeric) - Sentence Writing Fluency.

RDGREC (numeric) - Reading Recall.

NUMMAT (numeric) - Number Matrices.

EDIT (numeric) - Editing.

WRDFLU (numeric) - Word Reading Fluency.

SPLSND (numeric) - Spelling of Sounds.

RDGVOC (numeric) - Reading Vocabulary.

SCI (numeric) - Science.

SOC (numeric) - Social Studies.

HUM (numeric) - Humanities.

Source

McGrew, K. S., LaForte, E. M., & Schrank, F. A. (2014). Technical Manual. Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

`WJIV_ages_20_39` *Woodcock Johnson IV: ages 20 to 39*

Description

A list containing the bivariate correlations (N = 1,251) of the 47 intelligence subtests from the WJ IV for 20- to 39-year-olds obtained from the WJ-IV technical manual (McGrew, LaForte, & Schrank, 2014). Tables are reproduced with permission from the publisher.

Usage

```
WJIV_ages_20_39
```

Format

A list of 2 with elements "cormat" (47 x 47 matrix of bivariate correlations) and "N" (scalar). The correlation matrix contains the following variables:

ORLVOC (numeric) - Oral Vocabulary.

NUMSER (numeric) - Number Series.

VRBATN (numeric) - Verbal Attention.

LETPAT (numeric) - Letter-Pattern Matching.

PHNPRO (numeric) - Phonological Processing.

STYREC (numeric) - Story Recall.

VISUAL (numeric) - Visualization.

GENINF (numeric) - General Information.

CONFRM (numeric) - Concept Formation.

NUMREV (numeric) - Numbers Reversed.

NUMPAT (numeric) - Number-Pattern Matching.

NWDREP (numeric) - Nonword Repetition.

VAL (numeric) - Visual-Auditory Learning.

PICREC (numeric) - Picture Recognition.

ANLSYN (numeric) - Analysis-Synthesis.

OBJNUM (numeric) - Object-Number Sequencing.

PAIRCN (numeric) - Pair Cancellation.

MEMWRD (numeric) - Memory for Words.

PICVOC (numeric) - Picture Vocabulary.

ORLCMP (numeric) - Oral Comprehension.

SEGMNT (numeric) - Segmentation.

RPCNAM (numeric) - Rapid Picture Naming.

SENREP (numeric) - Sentence Repetition.

UNDDIR (numeric) - Understanding Directions.

SNDBLN (numeric) - Sound Blending.

RETFLU (numeric) - Retrieval Fluency.

SNDAWR (numeric) - Sound Awareness.

LWIDNT (numeric) - Letter-Word Identification.

APPROB (numeric) - Applied Problems.

SPELL (numeric) - Spelling.

PSGCMP (numeric) - Passage Comprehension.

CALC (numeric) - Calculation.

WRTSMP (numeric) - Writing Samples.

WRDATK (numeric) - Word Attack.

ORLRDG (numeric) - Oral Reading.

SNRDFL (numeric) - Sentence Reading Fluency.

MTHFLU (numeric) - Math Facts Fluency.

SNWRFL (numeric) - Sentence Writing Fluency.

RDGREC (numeric) - Reading Recall.

NUMMAT (numeric) - Number Matrices.

EDIT (numeric) - Editing.

WRDFLU (numeric) - Word Reading Fluency.

SPLSND (numeric) - Spelling of Sounds.

RDGVOC (numeric) - Reading Vocabulary.

SCI (numeric) - Science.

SOC (numeric) - Social Studies.

HUM (numeric) - Humanities.

Source

McGrew, K. S., LaForte, E. M., & Schrank, F. A. (2014). Technical Manual. Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

WJIV_ages_3_5   *Woodcock Johnson IV: ages 3 to 5*

Description

A list containing the bivariate correlations (N = 435) of the 29 intelligence subtests from the WJ IV for 3- to 5-year-olds obtained from the WJ IV technical Manual (McGrew, LaForte, & Schrank, 2014). Tables are reproduced with permission from the publisher.

Usage

```
WJIV_ages_3_5
```

Format

A list of 2 with elements "cormat" (29 x 29 matrix of bivariate correlations) and "N" (scalar). The correlation matrix contains the following variables:

ORLVOC (numeric) - Oral Vocabulary.

VRBATN (numeric) - Verbal Attention.

LETPAT (numeric) - Phonological Processing.

STYREC (numeric) - Story Recall.

VISUAL (numeric) - Visualization.

GENINF (numeric) - General Information.

CONFRM (numeric) - Concept Formation.

NUMREV (numeric) - Numbers Reversed.

NUMPAT (numeric) - Number-Pattern Matching.

NWDREP (numeric) - Nonword Repetition.

VAL (numeric) - Visual-Auditory Learning.

PICREC (numeric) - Picture Recognition.

MEMWRD (numeric) - Memory for Words.

PICVOC (numeric) - Picture Vocabulary.

ORLCMP (numeric) - Oral Comprehension.

SEGMNT (numeric) - Segmentation.

RPCNAM (numeric) - Rapid Picture Naming.

SENREP (numeric) - Sentence Repetition.

UNDDIR (numeric) - Understanding Directions.

SNDBLN (numeric) - Sound Blending.

RETFLU (numeric) - Retrieval Fluency.

SNDAWR (numeric) - Sound Awareness.

LWIDNT (numeric) - Letter-Word Identification.

APPROB (numeric) - Applied Problems.

SPELL (numeric) - Spelling.

PSGCMP (numeric) - Passage Comprehension.

SCI (numeric) - Science.

SOC (numeric) - Social Studies.

HUM (numeric) - Humanities.

Source

McGrew, K. S., LaForte, E. M., & Schrank, F. A. (2014). Technical Manual. Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

`WJIV_ages_40_90` *Woodcock Johnson IV: ages 40 to 90 plus*

Description

A list containing the bivariate correlations (N = 1,146) of the 47 intelligence subtests from the WJ IV for 40- to 90+-year-olds obtained from the WJ-IV technical manual (McGrew, LaForte, & Schrank, 2014). Tables are reproduced with permission from the publisher.

Usage

```
WJIV_ages_40_90
```

Format

A list of 2 with elements "cormat" (47 x 47 matrix of bivariate correlations) and "N". The correlation matrix contains the following variables:

ORLVOC (numeric) - Oral Vocabulary.

NUMSER (numeric) - Number Series.

VRBATN (numeric) - Verbal Attention.

LETPAT (numeric) - Letter-Pattern Matching.

PHNPRO (numeric) - Phonological Processing.

STYREC (numeric) - Story Recall.

VISUAL (numeric) - Visualization.

GENINF (numeric) - General Information.

CONFRM (numeric) - Concept Formation.

NUMREV (numeric) - Numbers Reversed.

NUMPAT (numeric) - Number-Pattern Matching.

NWDREP (numeric) - Nonword Repetition.

VAL (numeric) - Visual-Auditory Learning.

PICREC (numeric) - Picture Recognition.

ANLSYN (numeric) - Analysis-Synthesis.

OBJNUM (numeric) - Object-Number Sequencing.

PAIRCN (numeric) - Pair Cancellation.

MEMWRD (numeric) - Memory for Words.

PICVOC (numeric) - Picture Vocabulary.

ORLCMP (numeric) - Oral Comprehension.

SEGMNT (numeric) - Segmentation.

RPCNAM (numeric) - Rapid Picture Naming.

SENREP (numeric) - Sentence Repetition.

UNDDIR (numeric) - Understanding Directions.

SNDBLN (numeric) - Sound Blending.

RETFLU (numeric) - Retrieval Fluency.

SNDAWR (numeric) - Sound Awareness.

LWIDNT (numeric) - Letter-Word Identification.

APPROB (numeric) - Applied Problems.

SPELL (numeric) - Spelling.

PSGCMP (numeric) - Passage Comprehension.

CALC (numeric) - Calculation.

WRTSMP (numeric) - Writing Samples.

WRDATK (numeric) - Word Attack.

ORLRDG (numeric) - Oral Reading.

SNRDFL (numeric) - Sentence Reading Fluency.

MTHFLU (numeric) - Math Facts Fluency.

SNWRFL (numeric) - Sentence Writing Fluency.

RDGREC (numeric) - Reading Recall.

NUMMAT (numeric) - Number Matrices.

EDIT (numeric) - Editing.

WRDFLU (numeric) - Word Reading Fluency.

SPLSND (numeric) - Spelling of Sounds.

RDGVOC (numeric) - Reading Vocabulary.

SCI (numeric) - Science.

SOC (numeric) - Social Studies.

HUM (numeric) - Humanities.

Source

McGrew, K. S., LaForte, E. M., & Schrank, F. A. (2014). Technical Manual. Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

`WJIV_ages_6_8` *Woodcock Johnson IV: ages 6 to 8*

Description

A list containing the bivariate correlations (N = 825) of the 47 intelligence subtests from the WJ IV for 6- to 8-year-olds obtained from the WJ-IV technical manual (McGrew, LaForte, & Schrank, 2014). Tables are reproduced with permission from the publisher.

Usage

```
WJIV_ages_6_8
```

Format

A list of 2 with elements "cormat" (47 x 47 matrix of bivariate correlations) and "N". The correlation matrix contains the following variables:

ORLVOC (numeric) - Oral Vocabulary.

NUMSER (numeric) - Number Series.

VRBATN (numeric) - Verbal Attention.

LETPAT (numeric) - Letter-Pattern Matching.

PHNPRO (numeric) - Phonological Processing.

STYREC (numeric) - Story Recall.

VISUAL (numeric) - Visualization.

GENINF (numeric) - General Information.

CONFRM (numeric) - Concept Formation.

NUMREV (numeric) - Numbers Reversed.

NUMPAT (numeric) - Number-Pattern Matching.

NWDREP (numeric) - Nonword Repetition.

VAL (numeric) - Visual-Auditory Learning.

PICREC (numeric) - Picture Recognition.

ANLSYN (numeric) - Analysis-Synthesis.

OBJNUM (numeric) - Object-Number Sequencing.

PAIRCN (numeric) - Pair Cancellation.

MEMWRD (numeric) - Memory for Words.

PICVOC (numeric) - Picture Vocabulary.

ORLCMP (numeric) - Oral Comprehension.

SEGMNT (numeric) - Segmentation.

RPCNAM (numeric) - Rapid Picture Naming.

SENREP (numeric) - Sentence Repetition.

UNDDIR (numeric) - Understanding Directions.

SNDBLN (numeric) - Sound Blending.

RETFLU (numeric) - Retrieval Fluency.

SNDAWR (numeric) - Sound Awareness.

LWIDNT (numeric) - Letter-Word Identification.

APPROB (numeric) - Applied Problems.

SPELL (numeric) - Spelling.

PSGCMP (numeric) - Passage Comprehension.

CALC (numeric) - Calculation.

WRTSMP (numeric) - Writing Samples.

WRDATK (numeric) - Word Attack.

ORLRDG (numeric) - Oral Reading.

SNRDFL (numeric) - Sentence Reading Fluency.

MTHFLU (numeric) - Math Facts Fluency.

SNWRFL (numeric) - Sentence Writing Fluency.

RDGREC (numeric) - Reading Recall.

NUMMAT (numeric) - Number Matrices.

EDIT (numeric) - Editing.

WRDFLU (numeric) - Word Reading Fluency.

SPLSND (numeric) - Spelling of Sounds.

RDGVOC (numeric) - Reading Vocabulary.

SCI (numeric) - Science.

SOC (numeric) - Social Studies.

HUM (numeric) - Humanities.

Source

McGrew, K. S., LaForte, E. M., & Schrank, F. A. (2014). Technical Manual. Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

WJIV_ages_9_13 *Woodcock Johnson IV: ages 9 to 13*

Description

A list containing the bivariate correlations (N = 1,572) of the 47 intelligence subtests from the WJ IV for 9- to 13-year-olds obtained from the WJ-IV technical manual (McGrew, LaForte, & Schrank, 2014). Tables are reproduced with permission from the publisher.

Usage

```
WJIV_ages_9_13
```

Format

A list of 2 with elements "cormat" (47 x 47 matrix of bivariate correlations) and "N". The correlation matrix contains the following variables:

ORLVOC (numeric) - Oral Vocabulary.

NUMSER (numeric) - Number Series.

VRBATN (numeric) - Verbal Attention.

LETPAT (numeric) - Letter-Pattern Matching.

PHNPRO (numeric) - Phonological Processing.

STYREC (numeric) - Story Recall.

VISUAL (numeric) - Visualization.

GENINF (numeric) - General Information.

CONFRM (numeric) - Concept Formation.

NUMREV (numeric) - Numbers Reversed.

NUMPAT (numeric) - Number-Pattern Matching.

NWDREP (numeric) - Nonword Repetition.

VAL (numeric) - Visual-Auditory Learning.

PICREC (numeric) - Picture Recognition.

ANLSYN (numeric) - Analysis-Synthesis.

OBJNUM (numeric) - Object-Number Sequencing.

PAIRCN (numeric) - Pair Cancellation.

MEMWRD (numeric) - Memory for Words.

PICVOC (numeric) - Picture Vocabulary.

ORLCMP (numeric) - Oral Comprehension.

SEGMNT (numeric) - Segmentation.

RPCNAM (numeric) - Rapid Picture Naming.

SENREP (numeric) - Sentence Repetition.

UNDDIR (numeric) - Understanding Directions.

SNDBLN (numeric) - Sound Blending.

RETFLU (numeric) - Retrieval Fluency.

SNDAWR (numeric) - Sound Awareness.

LWIDNT (numeric) - Letter-Word Identification.

APPROB (numeric) - Applied Problems.

SPELL (numeric) - Spelling.

PSGCMP (numeric) - Passage Comprehension.

CALC (numeric) - Calculation.

WRTSMP (numeric) - Writing Samples.

WRDATK (numeric) - Word Attack.

ORLRDG (numeric) - Oral Reading.

SNRDFL (numeric) - Sentence Reading Fluency.

MTHFLU (numeric) - Math Facts Fluency.

SNWRFL (numeric) - Sentence Writing Fluency.

RDGREC (numeric) - Reading Recall.

NUMMAT (numeric) - Number Matrices.

EDIT (numeric) - Editing.

WRDFLU (numeric) - Word Reading Fluency.

SPLSND (numeric) - Spelling of Sounds.

RDGVOC (numeric) - Reading Vocabulary.

SCI (numeric) - Science.

SOC (numeric) - Social Studies.

HUM (numeric) - Humanities.

Source

McGrew, K. S., LaForte, E. M., & Schrank, F. A. (2014). Technical Manual. Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). Woodcock-Johnson IV. Rolling Meadows, IL: Riverside.

Index