Package ‘EGAnet’

January 19, 2020

Title Exploratory Graph Analysis - A Framework for Estimating the Number of Dimensions in Multivariate Data Using Network Psychometrics

Version 0.9.0

Date 2020-1-18

Maintainer Hudson Golino <hfg9s@virginia.edu>

Description An implementation of the Exploratory Graph Analysis (EGA) framework for dimensionality assessment. EGA is part of a new area called network psychometrics that focuses on the estimation of undirected network models in psychological datasets. EGA estimates the number of dimensions or factors using graphical lasso or Triangulated Maximally Filtered Graph (TMFG) and a weighted network community analysis. A bootstrap method for verifying the stability of the estimation is also available. The fit of the structure suggested by EGA can be verified using confirmatory factor analysis and a direct way to convert the EGA structure to a confirmatory factor model is also implemented. Documentation and examples are available. Golino, H. F., & Epskamp, S. (2017) <doi:10.1371/journal.pone.0174035>. Golino, H. F., & Demetriou, A. (2017) <doi:10.1016/j.intell.2017.0
rido, L. E., Christensen, A. P., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (2018) <doi:10.31234/osf.io/gzcre>. Christensen, A. P. & Golino, H.F. (2019) <doi:10.31234/osf.io/9deay>.

Depends R (>= 3.5.0)

License GPL (>= 3.0)

Encoding UTF-8

LazyData true

Imports qgraph (>= 1.4.1), semPlot (>= 1.0.1), igraph (>= 1.0.1), lavaan (>= 0.5-22), doParallel (>= 1.0.10), foreach (>= 1.4.3), NetworkToolbox (>= 1.1.2), glasso (>= 1.10), dplyr (>= 0.7.8), Matrix (>= 1.2), plotly (>= 4.7.1), mvtnorm (>= 1.0.8), corpcor (>= 1.6.9), ggpubr (>= 0.2), iterators (>= 1.0.10), stats, plyr (>= 1.8.4), matrixcalc (>= 1.0-3), ggplot2 (>= 3.1.0), pbapply, OpenMx, wTO, psych (>= 1.8.10), MASS, fdrtool, fitdistrplus, tidyselect (<= 0.2.5), psychTools (>= 1.9.12)

Suggests knitr, rmarkdown, kableExtra

VignetteBuilder knitr

RoxygenNote 7.0.2

NeedsCompilation no

Author Hudson Golino [aut, cre] (<https://orcid.org/0000-0002-1601-1447>),
Alexander Christensen [aut] (<https://orcid.org/0000-0002-9798-7037>),
Robert Moulder [ctb] (<https://orcid.org/0000-0001-7504-9560>)

Repository CRAN

Date/Publication 2020-01-19 00:20:03 UTC

R topics documented:

EGAnet-package *EGAnet–package*

Description

An implementation of the Exploratory Graph Analysis (EGA) framework for dimensionality assessment. EGA is part of a new area called network psychometrics that focuses on the estimation of undirected network models in psychological datasets. EGA estimates the number of dimensions or factors using graphical lasso or Triangulated Maximally Filtered Graph (TMFG) and a weighted network community analysis. A bootstrap method for verifying the stability of the estimation is also available. The fit of the structure suggested by EGA can be verified using confirmatory factor analysis and a direct way to convert the EGA structure to a confirmatory factor model is also implemented. Documentation and examples are available.

Author(s)

Hudson Golino <hfg9s@virginia.edu>

References

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS one*, *12(6)*, e0174035.. doi: journal.pone.0174035

Golino, H. F., & Demetriou, A. (2017). Estimating the dimensionality of intelligence like data using Exploratory Graph Analysis. *Intelligence*, *62*, 54-70. doi: j.intell.2017.02.007

Golino, H., Shi, D., Garrido, L. E., Christensen, A. P., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (2018). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *PsyArXiv*. doi: 10.31234/osf.io/gzcre

boot.wmt `bootEGA` *Results of* `wmt2`*Data*

Description

`bootEGA` results using the `"glasso"` model and 500 iterations of the Wiener Matrizen-Test 2 (WMT-2)

Usage

```
data(boot.wmt)
```

Format

A list with 8 objects (see `bootEGA`)

Examples

```
data("boot.wmt")
```

---

bootEGA *Dimension Stability Analysis of* EGA

Description

bootEGA Estimates the number of dimensions of *n* bootstraps using the empirical (partial) correlation matrix (parametric) or resampling from the empirical dataset (non-parametric). It also estimates a typical median network structure, which is formed by the median or mean pairwise (partial) correlations over the *n* bootstraps.

Usage

```
bootEGA(
  data,
  n,
  model = c("glasso", "TMFG"),
  type = c("parametric", "resampling"),
  typicalStructure = TRUE,
  plot.typicalStructure = TRUE,
  ncores = 4,
  ...
)
```

Arguments

data — Matrix or data frame. Includes the variables to be used in the bootEGA analysis

n — Numeric integer. Number of replica samples to generate from the bootstrap analysis. At least 500 is recommended

model — Character. A string indicating the method to use. Defaults to "glasso".

Current options are:

- "glasso" Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter. See EBICglasso.qgraph
- "TMFG" Estimates a Triangulated Maximally Filtered Graph. See TMFG

type — Character. A string indicating the type of bootstrap to use.

Current options are:

- "parametric" Generates n new datasets (multivariate normal random distributions) based on the original dataset, via the Mvnorm function of the mvtnorm package
- "resampling" Generates n random subsamples of the original data

typicalStructure
: Boolean. If TRUE, returns the typical network of partial correlations (estimated via graphical lasso or via TMFG) and estimates its dimensions. The "typical network" is the median of all pairwise correlations over the *n* bootstraps. Defaults to TRUE

plot.typicalStructure
: Boolean. If TRUE, returns a plot of the typical network (partial correlations), which is the median of all pairwise correlations over the *n* bootstraps, and its estimated dimensions. Defaults to TRUE

ncores
: Numeric. Number of cores to use in computing results. Defaults to 4. Set to 1 to not use parallel computing. Recommended to use maximum number of cores minus one

  If you're unsure how many cores your computer has, then use the following code: `parallel::detectCores()`

...
: Additional arguments to be passed to EBICglasso.qgraph or TMFG

Value

Returns a list containing:

n
: Number of replica samples in bootstrap

boot.ndim
: Number of dimensions identified in each replica sample

boot.wc
: Item allocation for each replica sample

bootGraphs
: Networks of each replica sample

summary.table
: Summary table containing number of replica samples, median, standard deviation, standard error, 95% confidence intervals, and quantiles (lower = 2.5% and upper = 97.5%)

frequency
: Proportion of times the number of dimensions was identified (e.g., .85 of 1,000 = 850 times that specific number of dimensions was found)

EGA
: Output of the original EGA results

typicalGraph
: A list containing:

  - graph Network matrix of the median network structure
  - typical.dim.variables An ordered matrix of item allocation
  - wc Item allocation of the median network

Author(s)

Hudson F. Golino <hfg9s at virginia.edu> and Alexander P. Christensen <alexpaulchristensen@gmail.com>

References

Christensen, A. P., & Golino, H. F. (2019). Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis: A tutorial. *PsyArXiv*. doi:10.31234/osf.io/9deay

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:

# bootEGA glasso example
boot.wmt <- bootEGA(data = wmt, n = 500, typicalStructure = TRUE,
plot.typicalStructure = TRUE, model = "glasso", type = "parametric", ncores = 4)

## End(Not run)

# Load data
intwl <- intelligenceBattery[,8:66]

## Not run:
# bootEGA TMFG example
boot.intwl <- bootEGA(data = intelligenceBattery[,8:66], n = 500, typicalStructure = TRUE,
plot.typicalStructure = TRUE, model = "TMFG", type = "parametric", ncores = 4)


## End(Not run)
```

CFA *CFA Fit of EGA Structure*

Description

Verifies the fit of the structure suggested by EGA using confirmatory factor analysis

Usage

```
CFA(ega.obj, data, estimator, plot.CFA = TRUE, layout = "spring", ...)
```

Arguments

ega.obj	An EGA object
data	A dataframe with the variables to be used in the analysis
estimator	The estimator used in the confirmatory factor analysis. 'WLSMV' is the estimator of choice for ordinal variables. 'ML' or 'WLS' for interval variables. See lavOptions for more details

plot.CFA	Logical. Should the CFA structure with its standardized loadings be plot? Defaults to TRUE
layout	Layout of plot (see semPaths). Defaults to "spring"
...	Arguments passed to cfa

Value

Returns a list containing:

fit	Output from cfa
summary	Summary output from lavaan-class
fit.measures	Fit measures: chi-squared, degrees of freedom, p-value, CFI, RMSEA, GFI, and NFI. Additional fit measures can be applied using the fitMeasures function (see examples)

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and bootEGA to investigate the stability of EGA's estimation via bootstrap.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate EGA
ega.wmt <- EGA(data = wmt)


## End(Not run)

# Fit CFA model to EGA results
cfa.wmt <- CFA(ega.obj = ega.wmt, estimator = 'WLSMV', plot.CFA = TRUE, data = wmt)

# Additional fit measures
lavaan::fitMeasures(cfa.wmt$fit, fit.measures = "all")

# Load data
intel <- intelligenceBattery[,8:66]

## Not run:
# Estimate EGA
ega.intel <- EGA(data = intel)
```

```
# Fit CFA model to EGA results
cfa.intel <- CFA(ega.obj = ega.intel, estimator = 'WLSMV', plot.CFA = TRUE,
data = intel)


## End(Not run)
```

---

cmi *Conditional Mutual Information*

Description

Computes the conditional mutual information metric using a modification of the matrix of partial correlations (see Zhao, Zhou,Zhang, & Chen, 2016). If the raw data is provided, the correlation matrix will be computed using the `cor_auto` function of the `qgraph` package.

Usage

```
cmi(data, network = FALSE, EGA = TRUE, steps = 4)
```

Arguments

`data`	A dataframe with the variables to be used in the analysis or a correlation matrix.
`network`	Logical. If TRUE, returns a plot of the conditional mutual information network. Defaults to FALSE.
`EGA`	Logical. If TRUE, exploratory graph analysis is performed using the conditional mutual information network.
`steps`	Number of steps to be used in `cluster_walktrap` algorithm (necessary only if the EGA argument is set to TRUE). Defaults to 4.

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>#'

References

Zhao, J., Zhou, Y., Zhang, X., & Chen, L. (2016). Part mutual information for quantifying direct associations in networks. *Proceedings of the National Academy of Sciences*, *113*, 5130-5135. doi: 10.1073/pnas.1522586113

See Also

`bootEGA` to investigate the stability of EGA's estimation via bootstrap and `EGA` to apply the exploratory graph analysis technique.

Examples

```
wmt <- wmt2[,7:24]


#estimate EGA
ega.wmt <- EGA(data = wmt, model = "glasso", plot.EGA = TRUE)


#estimate EGAtmfg
ega.wmt <- EGA(data = wmt, model = "TMFG", plot.EGA = TRUE)

#summary statistics
summary(ega.wmt)

#plot
plot(ega.wmt)

#estimate EGA
ega.intel <- EGA(data = intelligenceBattery[,8:66], model = "glasso", plot.EGA = TRUE)

#summary statistics
summary(ega.intel)

#plot
plot(ega.intel)
```

`depression` *Depression Data*

Description

A response matrix (n = 574) of the Beck Depression Inventory, Beck Anxiety Inventory and the Athens Insomnia Scale.

Usage

```
data(depression)
```

Format

A 574x78 response matrix

Examples

```
data("depression")
```

dimStability *Dimension Stability Statistics from* bootEGA

Description

Based on the bootEGA results, this function computes the stability of dimensions. This is computed by assessing the proportion of items that replicate within the defined factor/dimension (see argument orig.wc) for each bootstrap. The mean of these proportions represent the dimensional stability for each dimension

Usage

```
dimStability(bootega.obj, orig.wc, item.stability = TRUE)
```

Arguments

bootega.obj	A bootEGA object
orig.wc	Numeric or character. A vector with community numbers or labels for each item. Typically uses community results (wc) from EGA
item.stability	Boolean. Should the item stability statistics be computed using [EGAnet]{itemStability}? Defaults to TRUE

Value

When argument item.stability = TRUE, returns a list containing:

dimensions	The dimensional stability of each dimension
items	The output from [EGAnet]{itemStability}

When argument item.stability = FALSE, returns a vector of the dimensional stability of each dimension

Author(s)

Hudson F. Golino <hfg9s at virginia.edu> and Alexander P. Christensen <alexpaulchristensen@gmail.com>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate EGA network
ega.wmt <- EGA(data = wmt, model = "glasso")

# Estimate dimension stability
boot.wmt <- bootEGA(data = wmt, n = 100, typicalStructure = TRUE,
plot.typicalStructure = TRUE, model = "glasso",
type = "parametric", ncores = 4)


## End(Not run)

# Estimate item stability statistics
dimStability(boot.wmt, orig.wc = ega.wmt$wc, item.stability = FALSE)
```

---

dynamic.plot *Dynamic Plot method for EGA objects*

Description

Plots the EGA result using plotly

Usage

```
dynamic.plot(ega.obj, title = "", vsize = 30, opacity = 0.4)
```

Arguments

ega.obj	An EGA object
title	Character. Title of the plot
vsize	Numeric. An integer indicating the size of the nodes. Default vsize = 30
opacity	Numeric. A numeric value indicating the opacity of the edges. Default opacity = 0.4

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
#estimate EGA
ega.wmt <- EGA(data = wmt, plot.EGA = TRUE)


## End(Not run)

# Summary of EGA results
summary(ega.wmt)

## Not run:
# Dynamic plot
dynamic.plot(ega.wmt, title = "", vsize = 30, opacity = 0.4)


## End(Not run)
```

EBICglasso.qgraph   EBICglasso *from* qgraph *1.4.4*

Description

This function uses the glasso package (Friedman, Hastie and Tibshirani, 2011) to compute a sparse gaussian graphical model with the graphical lasso (Friedman, Hastie & Tibshirani, 2008). The tuning parameter is chosen using the Extended Bayesian Information criterium (EBIC) described by Foygel & Drton (2010).

Usage

```
EBICglasso.qgraph(
  data,
  n = NULL,
  gamma = 0.5,
  penalize.diagonal = FALSE,
  nlambda = 100,
  lambda.min.ratio = 0.01,
  returnAllResults = FALSE,
  penalizeMatrix,
  countDiagonal = FALSE,
  refit = FALSE,
  ...
)
```

Arguments

`data`	Data matrix
`n`	Number of participants
`gamma`	EBIC tuning parameter. 0.5 is generally a good choice. Setting to zero will cause regular BIC to be used.
`penalize.diagonal`	Should the diagonal be penalized?
`nlambda`	Number of lambda values to test.
`lambda.min.ratio`	Ratio of lowest lambda value compared to maximal lambda
`returnAllResults`	If `TRUE` this function does not return a network but the results of the entire glasso path.
`penalizeMatrix`	Optional logical matrix to indicate which elements are penalized
`countDiagonal`	Should diagonal be counted in EBIC computation? Defaults to `FALSE`. Set to `TRUE` to mimic qgraph < 1.3 behavior (not recommended!).
`refit`	Logical, should the optimal graph be refitted without LASSO regularization? Defaults to `FALSE`.
`...`	Arguments sent to `glasso`

Details

The glasso is run for 100 values of the tuning parameter logarithmically spaced between the maximal value of the tuning parameter at which all edges are zero, lambda_max, and lambda_max/100. For each of these graphs the EBIC is computed and the graph with the best EBIC is selected. The partial correlation matrix is computed using `wi2net` and returned.

Value

A partial correlation matrix

Author(s)

Sacha Epskamp <mail@sachaepskamp.com>

References

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*, 432-441. doi: 10.1093/biostatistics/kxm045

#glasso package Jerome Friedman, Trevor Hastie and Rob Tibshirani (2011). glasso: Graphical lasso-estimation of Gaussian graphical models. R package version 1.7. `https://CRAN.R-project.org/package=glasso`

Foygel, R., & Drton, M. (2010). Extended Bayesian information criteria for Gaussian graphical models. In Advances in neural information processing systems (pp. 604-612). `https://papers.nips.cc/paper/4087-extended-bayesian-information-criteria-for-gaussian-graphical-models`

#psych package Revelle, W. (2014) psych: Procedures for Personality and Psychological Research, Northwestern University, Evanston, Illinois, USA. R package version 1.4.4. https://CRAN.R-project.org/package=psych

#Matrix package Douglas Bates and Martin Maechler (2014). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.1-3. https://CRAN.R-project.org/package=Matrix

Examples

```
### Using wmt2 dataset from EGAnet ###
data(wmt2)

## Not run:
# Compute correlations:
CorMat <- cor_auto(wmt2[,7:24])

# Compute graph with tuning = 0 (BIC):
BICgraph <- EBICglasso.qgraph(CorMat, nrow(wmt2), 0)

# Compute graph with tuning = 0.5 (EBIC)
EBICgraph <- EBICglasso.qgraph(CorMat, nrow(wmt2), 0.5)


## End(Not run)
```

EGA *Apply the Exploratory Graph Analysis technique*

Description

Estimates the number of dimensions of a given dataset/instrument using graphical lasso (EBICglasso.qgraph) or the Triangulated Maximally Filtered Graph (TMFG) method and the walktrap community detection algorithm (cluster_walktrap). The glasso regularization parameter is set via EBIC.

Usage

```
EGA(
  data,
  model = c("glasso", "TMFG"),
  plot.EGA = TRUE,
  n = NULL,
  steps = 4,
  nvar = 4,
  nfact = 1,
  load = 0.7,
  ...
)
```

Arguments

`data`	A dataframe with the variables to be used in the analysis or a correlation matrix. If the data used is a correlation matrix, the argument `n` will need to be specified.
`model`	A string indicating the method to use. Current options are: • `glasso` Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter. This is the default method • `TMFG` Estimates a Triangulated Maximally Filtered Graph
`plot.EGA`	Logical. If TRUE, returns a plot of the network and its estimated dimensions. Defaults to TRUE
`n`	Integer. Sample size, if the data provided is a correlation matrix
`steps`	Number of steps to be used in `cluster_walktrap` algorithm. Defaults to 4.
`nvar`	Number of variables to use in the simulation part of the unidimensionality check. Defaults to 4.
`nfact`	Number of factors to be simulated (part of the unidimensionality check algorithm). Defaults to 1.
`load`	Factor loadings (used in the unidimensionality check algorithm). Defaults to 0.70.
`...`	Additional arguments to be passed to `EBICglasso.qgraph` or `TMFG`

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>, Alexander P. Christensen <alexpaulchristensen at gmail.com>, Maria Dolores Nieto <acinodam at gmail.com> and Luis E. Garrido <garrido.luiseduardo at gmail.com>

References

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS one*, *12(6)*, e0174035.. doi: journal.pone.0174035

Golino, H. F., & Demetriou, A. (2017). Estimating the dimensionality of intelligence like data using Exploratory Graph Analysis. *Intelligence*, *62*, 54-70. doi: j.intell.2017.02.007

Golino, H., Shi, D., Garrido, L. E., Christensen, A. P., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (2018). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *PsyArXiv*. doi: 10.31234/osf.io/gzcre

See Also

`bootEGA` to investigate the stability of EGA's estimation via bootstrap and `CFA` to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
#estimate EGA
ega.wmt <- EGA(data = wmt2[,7:24], model = "glasso", plot.EGA = TRUE)


#estimate EGAtmfg
ega.wmt <- EGA(data = wmt2[,7:24], model = "TMFG", plot.EGA = TRUE)

#summary statistics
summary(ega.wmt)

#plot
plot(ega.wmt)

#estimate EGA
ega.intel <- EGA(data = intelligenceBattery[,8:66], model = "glasso", plot.EGA = TRUE)

#summary statistics
summary(ega.intel)

#plot
plot(ega.intel)
```

EGA.fit — EGA *Optimal Model Fit using* entropyFit

Description

Estimates the best fitting model using EGA. The number of steps in the cluster_walktrap detection algorithm is varied and unique community solutions are compared using tefi.

Usage

```
EGA.fit(
  data,
  model = c("glasso", "TMFG"),
  steps = c(3, 4, 5, 6, 7, 8),
  n = NULL
)
```

Arguments

data	A dataset (or a correlation matrix).
model	Character. A string indicating the method to use. Defaults to "glasso".

steps — Numeric vector. Range of steps to be used in the model selection. Defaults from 3 to 8 steps (based on Pons & Latapy, 2006)

n — Integer. Sample size, if the data provided is a correlation matrix

Current options are:

- "glasso" Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter. See EBICglasso.qgraph
- "TMFG" Estimates a Triangulated Maximally Filtered Graph. See TMFG

Value

Returns a list containing:

EGA — The EGA output for the best fitting model

steps — The number of steps used in the best fitting model from the cluster_walktrap algorithm

EntropyFit — The tefi Index for the unique solutions given the range of steps (vector names represent the number of steps)

Lowest.EntropyFit — The lowest value for the tefi Index

Author(s)

Hudson F. Golino <hfg9s at virginia.edu> and Alexander P. Christensen <alexpaulchristensen@gmail.com>

References

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, *10*, 191-218. doi:10.7155/jgaa.00185

See Also

bootEGA to investigate the stability of EGA's estimation via bootstrap, EGA to estimate the number of dimensions of an instrument using EGA, and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate normal EGAtmfg
tmfg <- EGA(data = wmt, model = "TMFG")

# Estimate optimal EGAtmfg
tmfg.opt <- EGA.fit(data = wmt, model = "TMFG")
```

```
# Compare with CFA
cfa.tmfg <- CFA(tmfg, estimator = "WLSMV", data = wmt)
cfa.opt <- CFA(tmfg.opt$EGA, estimator = "WLSMV", data = wmt)

lavaan::lavTestLRT(cfa.tmfg$fit, cfa.opt$fit, method = "satorra.bentler.2001")


## End(Not run)
```

`ega.wmt` *EGA Network of wmt2Data*

Description

An EGA using the "glasso" model of the Wiener Matrizen-Test 2 (WMT-2)

Usage

```
data(ega.wmt)
```

Format

A 17 x 17 adjacency matrix

Examples

```
data("ega.wmt")
```

`entropyFit` *Entropy Fit Index*

Description

Computes the fit of a dimensionality structure using empirical entropy. Lower values suggest better fit of a structure to the data.

Usage

```
entropyFit(data, structure)
```

Arguments

`data`	Matrix or data frame. Contains variables to be used in the analysis
`structure`	A vector representing the structure (numbers or labels for each item). Can be theoretical factors or the structure detected by EGA

Value

Returns a list containing:

`Total.Correlation`
: The total correlation of the dataset

`Total.Correlation.MM`
: Miller-Madow correction for the total correlation of the dataset

`Entropy.Fit`
: The Entropy Fit Index

`Entropy.Fit.MM`
: Miller-Madow correction for the Entropy Fit Index

`Average.Entropy`
: The average entropy of the dataset

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>, Alexander P. Christensen <alexpaulchristensen@gmail.com> and Robert Moulder <rgm4fd@virginia.edu>

References

Golino, H. F., Moulder, R., Shi, D., Christensen, A. P., Neito, M. D., Nesselroade, J. R., & Boker, S. M. (under review) Entropy Fit Index: A new fit measure for assessing the structure and dimensionality of multiple latent variables. Retrieved from: https://www.researchgate.net/profile/Hudson_Golino/publication/333753928

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate EGA model
ega.wmt <- EGA(data = wmt, model = "glasso")


## End(Not run)

# Compute entropy indices
entropyFit(data = wmt, structure = ega.wmt$wc)
```

intelligenceBattery *Intelligence Data*

Description

A response matrix (n = 1152) of the International Cognitive Ability Resource (ICAR) intelligence battery developed by Condon and Revelle (2016).

Usage

```
data(intelligenceBattery)
```

Format

A 1185x125 response matrix

Examples

```
data("intelligenceBattery")
```

itemStability *Item Stability Statistics from bootEGA*

Description

Based on the bootEGA results, this function computes and plots the number of times an item (variable) is estimated in the same factor/dimension as originally estimated by EGA (`item.replication`). The output also contains each item's replication frequency (i.e., proportion of bootstraps that an item appeared in each dimension; `item.dim.rep`) as well as the average network loading for each item in each dimension (`item.loadings`).

Usage

```
itemStability(bootega.obj, orig.wc, item.freq = 0.1, plot.item.rep = TRUE)
```

Arguments

`bootega.obj`	A bootEGA object
`orig.wc`	Numeric or character. A vector with community numbers or labels for each item. Typically uses community results (`wc`) from EGA
`item.freq`	A value for lowest frequency allowed in `item.dim.rep` output. Removes noise from table to allow for easier interpretation. Defaults to `.10`
`plot.item.rep`	Should the plot be produced for `item.replication`? If `TRUE`, then a plot for the `item.replication` output will be produced.#' Defaults to `TRUE`

Value

Returns a list containing:

`item.replication`
: The proportion of times each item replicated within the defined dimension

`item.dim.rep`
: The proportion of times each item replicated within each possible dimension. Dimensions greater than the maximum number used in the `orig.wc` argument are labeled based on the largest remaining components after the dimensions used to `orig.wc`

`item.loadings`
: Matrix of the average standardized network loading (computed using `net.loads`) for each item in each dimension

`wc`
: A matrix containing the community membership values for each bootstrapped sample. The values correspond to the values input for the `orig.wc` argument

`plot.itemStability`
: A plot of the number of times each item replicates in its original community membership (`orig.wc`)

Author(s)

Hudson F. Golino <hfg9s at virginia.edu> and Alexander P. Christensen <alexpaulchristensen@gmail.com>

References

Danon, L., Diaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, *9*, P09008. <doi:10.1088/1742-5468/2005/09/P09008>

See Also

`EGA` to estimate the number of dimensions of an instrument using EGA and `CFA` to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate EGA network
ega.wmt <- EGA(data = wmt, model = "glasso")

# Estimate dimension stability
boot.wmt <- bootEGA(data = wmt, n = 100, typicalStructure = TRUE,
plot.typicalStructure = TRUE, model = "glasso",
type = "parametric", ncores = 4)


## End(Not run)
```

```
# Estimate item stability statistics
itemStability(boot.wmt, orig.wc = ega.wmt$wc)
```

---

net.loads *Network Loadings*

Description

Computes the between- and within-community strength of each item for each community. This function uses the comcat and stable functions to calculate the between- and within-community strength of each item, respectively.

Usage

```
net.loads(A, wc, rm.zero = FALSE, plot = FALSE)
```

Arguments

A	Matrix, data frame, or EGA object. An adjacency matrix of network data
wc	Numeric. A vector of community assignments. Not necessary if an EGA object is input for argument A
rm.zero	Should zeros be removed from the resulting matrix? Defaults to FALSE. Set to TRUE to reduce the noise in the results
plot	Boolean. Should proportional loadings be plotted? Defaults to FALSE. Set to TRUE for plot with pie charts visualizing the proportion of loading associated with each dimension

Details

Simulation studies have demonstrated that a node's strength centrality is roughly equivalent to factor loadings (Christensen, Golino, & Silvia, 2019; Hallquist, Wright, & Molenaar, in press). Hallquist and colleagues (in press) found that node strength represented a combination of dominant and cross-factor loadings. This function computes each node's strength within each specified dimension, providing a rough equivalent to factor loadings (including cross-loadings).

For more details, type `vignette("Network_Scores")`

Value

Returns a list containing:

unstd	A matrix of the unstandardized within- and between-community strength values for each node
std	A matrix of the standardized within- and between-community strength values for each node

Author(s)

Alexander P. Christensen <alexpaulchristensen@gmail.com> and Hudson F. Golino <hfg9s at virginia.edu>

References

Christensen, A. P., Golino, H. F., & Silvia, P. (2019). A psychometric network perspective on the measurement and assessment of personality traits. *PsyArXiv*. doi:10.31234/osf.io/ktejp

Hallquist, M., Wright, A. C. G., & Molenaar, P. C. (in press). Problems with centrality measures in psychopathology symptom networks: Why network psychometrics cannot escape psychometric theory. *Multivariate Behavioral Research*. doi:10.31234/osf.io/pg4mf

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate EGA
ega.wmt <- EGA(wmt)


## End(Not run)

# Network loadings
net.loads(ega.wmt, rm.zero = TRUE)
```

---

net.scores *Network Scores*

Description

This function computes network scores for factor analysis models. Network scores are computed based on each node's `strength` within each community (i.e., factor) in the network. These values are used as network "factor loadings" for the weights of each item. Notably, network analysis allows nodes to load onto more than one factor. These loadings are considered in the factor scores. In addition, if the construct is a hierarchy (e.g., personality questionnaire; items in facet scales in a trait domain), then an overall score can be computed (see argument `general`). These overall scores are computed using `comm.close` as weights, which are roughly similar to general factor loadings in a CFA model (see Christensen, Golino, & Silvia, 2019). The score estimates are roughly equivalent to the Maximum Likelihood method in `lavaan`'s `cfa` function. An important difference is that the network scores account for cross-loadings in their estimation of scores.

Usage

```
net.scores(data, A, wc, global = TRUE, type = c("sumscore", "latent"), ...)
```

Arguments

`data`	Matrix or data frame. Must be a dataset
`A`	Matrix, data frame, or `EGA` object. An adjacency matrix of network data
`wc`	Numeric. A vector of community assignments. Not necessary if an `EGA` object is input for argument `A`
`global`	Boolean. Should general network loadings be computed in scores? Defaults to `TRUE`. If there is more than one dimension and there is theoretically one global dimension, then general loadings of the dimensions onto the global dimension can be included in the weighted scores. For the type of weights (e.g., sum score or latent), see the `type` argument
`type`	Character. Should network scores parallel sum scores or latent variable scores? Defaults to `"latent"`. Argument `type` sets the community centrality measure that is used when computing the network loadings for multiple factors. Simulations have shown that `comm.eigen` computes weights that are closer to sum scores while `comm.close` computes weights that are closer to latent variable scores. See Christensen, Golino, and Silvia (2019) for more details
`...`	Additional arguments for `cluster_walktrap` and `louvain` community detection algorithms

Details

For more details, type `vignette("Network_Scores")`

Value

Returns a list containing:

`unstd.scores`	The unstandardized network scores for each participant and community (including the overall score)
`std.scores`	The standardized network scores for each participant and community (including the overall score)
`commCor`	Partial correlations between the specified or identified communities
`loads`	Standardized network loadings for each item in each dimension (computed using `net.loads`)

Author(s)

Alexander P. Christensen <alexpaulchristensen@gmail.com> and Hudson F. Golino <hfg9s at virginia.edu>

References

Christensen, A. P. (2018). NetworkToolbox: Methods and measures for brain, cognitive, and psychometric network analysis in R. *The R Journal*, *10*, 422-439. doi: 10.32614/RJ-2018-065

Christensen, A. P., Golino, H. F., & Silvia, P. J. (2019). A psychometric network perspective on the measurement and assessment of personality traits. *PsyArXiv*. doi: 10.31234/osf.io/ktejp

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
 # Estimate EGA
 ega.wmt <- EGA(wmt)

## End(Not run)

 # Network scores
 net.scores(data = wmt, A = ega.wmt)
```

---

node.redundant *Detects Redundant Nodes in a Network*

Description

Identifies redundant nodes in the network based on several measures. Computes the weighted topological overlap between each node and every other node in the network. The weighted topological overlap is implemented using the method from Nowick et al. (2009; see references) and the function wTO from the wTO package.

Usage

```
node.redundant(
  data,
  sig,
  type = c("wTO", "pcor", "thresh"),
  method = c("alpha", "bonferroni", "FDR", "adapt")
)
```

Arguments

data	Matrix or data frame
sig	Numeric. *p*-value for significance of overlap (defaults to `.05`). If more than 200 connections, then `fdrtool` is used to correct for false positives. In these instances, `sig` sets the *q*-value for significance of overlap (defaults to `.10`)
type	Character. Computes weighted topological overlap (`"wTO"` using `EBICglasso`), partial correlations (`"pcor"`), or thresholding based on a certain level of partial correlations (`"thresh"`). `type = "thresh"` will use the argument `"sig"` to input the desired threshold (defaults to `sig = .20`).
method	Character. Computes significance using the standard *p*-value (`"alpha"`), bonferroni corrected *p*-value (`"bonferroni"`), false-discovery rate corrected *p*-value (`"FDR"`), or adaptive alpha *p*-value (`adapt.a`). Defaults to `"alpha"`

Value

Returns a list:

redundant	Vectors nested within the list corresponding to redundant nodes with the name of object in the list
data	Returns original data
weights	Returns weights determine by weighted topological overlap or partial correlations
network	The network compute by EBICglasso

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

#wTO Nowick, K., Gernat, T., Almaas, E., & Stubbs, L. (2009). Differences in human and chimpanzee gene expression patterns define an evolving network of transcription factors in brain. *Proceedings of the National Academy of Sciences*, *106*, 22358-22363.

Examples

```
# obtain SAPA items
items <- psychTools::spi[,-c(1:10)]


# weighted topological overlap
redund <- node.redundant(items, type = "wTO", method = "adapt")

# partial correlation
redund <- node.redundant(items, type = "pcor", method = "adapt")
```

---

node.redundant.combine

Combines Redundant Nodes

Description

Allows user to combine redundant nodes into sum scores and latent variables to reduce the redundancy of variables in their data

Usage

```
node.redundant.combine(
  node.redundant.obj,
  type = c("sum", "optimal"),
  estimator = "WLSMV",
  auto = FALSE,
  ...
)
```

Arguments

`node.redundant.obj`
: A `node.redundant` object

`type`
: Character. Method to use to combine redundant variables.
  - `"sum"` Computes sum scores (i.e., means) of the variables
  - `"optimal"` Computes latent variables using `[lavaan]{cfa}` when there are more than 2 variables that are redundant. Computes sum scores (i.e., means) when there are 2 redundant variables

  Defaults to `"optimal"`

`estimator`
: Character. Estimator to use for latent variables. Defaults to `"WLSMV"`. See `[lavaan]{cfa}` for more options

`auto`
: NOT RECOMMENDED. Boolean. Should redundant nodes be automatically combined? Defaults to `FALSE`. If set to `TRUE`, then redundant nodes will combined using the following heuristics:

  1. Redundant nodes that form a 3-clique (i.e., a triangle) with the target node are automatically redundant

  2. If there are no 3-cliques, then the 2-clique with the largest regularized partial correlation is selected

`...`
: Options to be passed onto `[lavaan]{cfa}`

Value

Returns a list:

`data`
: New data with redundant variables merged

`merged`
: A matrix containing the variables that were decided to be redundant with one another

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# obtain SAPA items
items <- psychTools::spi[,-c(1:10)]


# weighted topological overlap
redund <- node.redundant(items, type = "wTO", method = "adapt")

# partial correlation
redund <- node.redundant(items, type = "pcor", method = "adapt")

# check redundancies
key.ind <- match(colnames(items), as.character(psychTools::spi.dictionary$item_id))
key <- as.character(psychTools::spi.dictionary$item[key.ind])

# change names in redundancy output to questionnaire item description
named.nr <- node.redundant.names(redund, key)


if(interactive())
{combine <- node.redundant.combine(named.nr)}
```

`node.redundant.names` *Changes Variable Names to Descriptions for* `node.redundant` *Objects*

Description

Using a key, this function changes the variable names in the `node.redundant` output to descriptions

Usage

```
node.redundant.names(node.redundant.obj, key)
```

Arguments

`node.redundant.obj`
: A `node.redundant` object

`key`
: Character vector. A vector with variable descriptions that correspond to the order of variables from the data used as input into the `node.redundant` function

Value

Returns a list:

`redundant`
: Vectors nested within the list corresponding to redundant nodes with the name of object in the list

`data` Returns original data

`weights` Returns weights determine by weighted topological overlap or partial correlations

`key` Returns original key

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# obtain SAPA items
items <- psychTools::spi[,-c(1:10)]


# weighted topological overlap
redund <- node.redundant(items, type = "wTO", method = "adapt")

# partial correlation
redund <- node.redundant(items, type = "pcor", method = "adapt")

# check redundancies
key.ind <- match(colnames(items), as.character(psychTools::spi.dictionary$item_id))
key <- as.character(psychTools::spi.dictionary$item[key.ind])

# change names in redundancy output to questionnaire item description
named.nr <- node.redundant.names(redund, key)
```

`optimism` *Optimism Data*

Description

A response matrix (n = 282) containing responses to 10 items of the Revised Life Orientation Test (LOT-R), developed by Scheier, Carver, & Bridges (1994).

Usage

```
data(optimism)
```

Format

A 282x10 response matrix

References

Scheier, M. F., Carver, C. S., & Bridges, M. W. (1994). Distinguishing optimism from neuroticism (and trait anxiety, self-mastery, and self-esteem): a reevaluation of the Life Orientation Test. *Journal of Personality and Social Psychology*, *67*, 1063-1078. doi: 10.1037//0022-3514.67.6.1063

Examples

```
data("optimism")
```

---

plot.bootEGA *Plot method for* bootEGA *objects*

Description

Plots bootEGA typical structure using qgraph

Usage

```
## S3 method for class 'bootEGA'
plot(x, vsize = 6, ...)
```

Arguments

x	A bootEGA object
vsize	An integer indicating the size of the nodes. Default vsize = 6
...	Arguments passed to qgraph

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
## Not run:
# Estimate EGA
ega.wmt <- EGA(data = wmt2[,7:24], model = "glasso")

# Estimate bootEGA
boot.wmt <- bootEGA(data = wmt2[,7:24], n = 10, typicalStructure = TRUE,
plot.typicalStructure = TRUE, model = "GGM",
type = "parametric", ncores = 4, confirm = ega.wmt$wc)
```

```
## End(Not run)

# Plot bootEGA
plot(boot.wmt)
```

---

plot.CFA *Plot Method for* CFA

Description

Plots the CFA structure using semPlot

Usage

```
## S3 method for class 'CFA'
plot(x, layout = "spring", vsize = 6, ...)
```

Arguments

x	An CFA object
layout	Layout of plot (see semPaths). Defaults to "spring"
vsize	Size of objects in plot. Defaults to 6
...	Arguments passed to semPaths in semPlot

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and bootEGA to investigate the stability of EGA's estimation via bootstrap.

Examples

```
## Not run:
# Estimate EGA
ega.wmt <- EGA(data = wmt2[,7:24], plot.EGA = TRUE)


## End(Not run)

# Estimate CFA
cfa.wmt <- CFA(ega.obj = ega.wmt, estimator = 'WLSMV', plot.CFA = FALSE, data = wmt2)

# Plot CFA
plot(cfa.wmt)
```

plot.EGA *Plot method for* EGA*objects*

Description

Plots the EGA result using qgraph

Usage

```
## S3 method for class 'EGA'
plot(x, title = "", vsize = 6, ...)
```

Arguments

x	An EGA object
title	Character. Title of the plot
vsize	An integer indicating the size of the nodes. Default vsize = 6
...	Arguments passed to qgraph

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
## Not run:
# Estimate EGA
ega.wmt <- EGA(data = wmt2[,7:24], plot.EGA = TRUE)


## End(Not run)

# Summary of EGA results
summary(ega.wmt)

# Plot EGA network
plot(ega.wmt, vsize = 6, label.prop = 1)
```

print.EGA *Print method for EGA objects*

Description

Returns a summary of the EGA results

Usage

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

Arguments

x	An EGA object
...	potentially further arguments (**unused currently**)

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
## Not run:
# Estimate EGA
ega.wmt <- EGA(data = wmt2[,7:24], plot.EGA = TRUE)


## End(Not run)

# Print EGA results
print(ega.wmt)
```

residualEGA *Residualized EGA*

Description

residualEGA Estimates the number of dimensions after controlling for wording effects. EGA is applied in the residual of a random intercept item factor model (RIIFA) with one method factor and one substantive factor.

Usage

```
residualEGA(data, manifests, lat, negative.items, plot = TRUE)
```

Arguments

data	Matrix or data frame. Includes the variables to be used in the residualEGA analysis
manifests	Character vector. Vector indicating the names of the variables (items) to be used in the analysis.
lat	Numeric integer. Number of latent factors to be estimated. Only one substantive latent factor is recommended in the current version of the function.
negative.items	Numeric vector A numeric vector indicating the column of the negative items.
plot	Boolean. If TRUE, returns a plot of the residualized network and its estimated dimensions. Defaults to TRUE

Value

Returns a list containing:

openMx.model	OpenMX model
openMx.result	OpenMX results
openMx.std.par	OpenMX standardized parameters
ResidualMatrix	Residual matrix
EGA.Residuals	Results of the residualized EGA
Fit	Fit metrics of the network structure, calculated using the ggmfit function of the qgraph package
WordLoads	Loadings of the wording effects

Author(s)

Hudson F. Golino <hfg9s at virginia.edu> and Robert Moulder <rgm4fd@virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
data <- optimism

## Not run:
# resEGA example
opt.res <- residualEGA(data = data, manifests = colnames(optimism),
lat = 1, negative.items = c(3,7,9), plot = TRUE)

# Fit:
opt.res$Fit

## End(Not run)
```

`summary.CFA` *Summary for CFA objects of `EGA` results*

Description

Returns a summary of the CFA results of `EGA` results

Usage

```
## S3 method for class 'CFA'
summary(object, ...)
```

Arguments

`object`	An `CFA` object
`...`	potentially further arguments (**unused currently**)

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

`EGA` to estimate the number of dimensions of an instrument using EGA and `bootEGA` to investigate the stability of EGA's estimation via bootstrap.

Examples

```
## Not run:
# Estimate EGA
ega.wmt <- EGA(data = wmt2[,7:24], plot.EGA = TRUE)
```

```
## End(Not run)

# Esimtate CFA
cfa.wmt <- CFA(ega.obj = ega.wmt, estimator = 'WLSMV', plot.CFA = TRUE, data = wmt2)

# Summary of CFA results
summary(cfa.wmt)
```

summary.EGA *Summary for* EGA *objects*

Description

Returns a summary of the EGA results

Usage

```
## S3 method for class 'EGA'
summary(object, ...)
```

Arguments

object	An EGA object
...	potentially further arguments (**unused currently**)

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

See Also

EGA to estimate the number of dimensions of an instrument using EGA and CFA to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
## Not run:
# Estimate EGA
ega.wmt <- EGA(data = wmt2[,7:24], plot.EGA = TRUE)


## End(Not run)

#Summary of EGA reults
summary(ega.wmt)
```

tefi — *Total Entropy Fit Index using Von Neumman's entropy (Quantum Information Theory) for correlation matrices*

Description

Computes the fit (TEFI) of a dimensionality structure using Von Neumman's entropy when the input is a correlation matrix. Lower values suggest better fit of a structure to the data.

Usage

```
tefi(data, structure)
```

Arguments

`data`	A dataset or a correlation matrix
`structure`	A vector representing the structure (numbers or labels for each item). Can be theoretical factors or the structure detected by `EGA`

Value

Returns a list containing:

`VN.Entropy.Fit`	The Entropy Fit Index using Von Neumman's entropy
`Total.Correlation`	The total correlation of the dataset
`Average.Entropy`	The average entropy of the dataset

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>, Alexander P. Christensen <alexpaulchristensen@gmail.com> and Robert Moulder <rgm4fd@virginia.edu>

See Also

`EGA` to estimate the number of dimensions of an instrument using EGA and `CFA` to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate EGA model
ega.wmt <- EGA(data = wmt, model = "glasso")
```

```
## End(Not run)

# Compute entropy indices
tefi(data = wmt, structure = ega.wmt$wc)
```

toy.example *Toy Example*

Description

A simulated dataset with 2 factors, three items per factor and n = 500.

Usage

```
data(toy.example)
```

Format

A 500x6 response matrix

Examples

```
data("toy.example")
```

vn.entropy *Entropy Fit Index using Von Neumman's entropy (Quantum Information Theory) for correlation matrices*

Description

Computes the fit of a dimensionality structure using Von Neumman's entropy when the input is a correlation matrix. Lower values suggest better fit of a structure to the data.

Usage

```
vn.entropy(data, structure)
```

Arguments

data	A dataset or a correlation matrix
structure	A vector representing the structure (numbers or labels for each item). Can be theoretical factors or the structure detected by EGA

Value

Returns a list containing:

`VN.Entropy.Fit` The Entropy Fit Index using Von Neumman's entropy

`Total.Correlation`
The total correlation of the dataset

`Average.Entropy`
The average entropy of the dataset

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>, Alexander P. Christensen <alexpaulchristensen@gmail.com> and Robert Moulder <rgm4fd@virginia.edu>

See Also

`EGA` to estimate the number of dimensions of an instrument using EGA and `CFA` to verify the fit of the structure suggested by EGA using confirmatory factor analysis.

Examples

```
# Load data
wmt <- wmt2[,7:24]

## Not run:
# Estimate EGA model
ega.wmt <- EGA(data = wmt, model = "glasso")


## End(Not run)

# Compute entropy indices
vn.entropy(data = wmt, structure = ega.wmt$wc)
```

---

wmt2 — *WMT-2 Data*

Description

A response matrix (n = 1185) of the Wiener Matrizen-Test 2 (WMT-2).

Usage

```
data(wmt2)
```

Format

A 1185x24 response matrix

Examples

```
data("wmt2")
```

Index