

Package ‘EGAnet’

July 13, 2020

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

Version 0.9.6

Date 2020-07-12

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.2017.09.001>. Golino, H. F., Christensen, A. P., Nieto, M. D., Sadana, R., & Thiagarajan, 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), NetworkToolbox (>= 1.1.2), glasso (>= 1.10), stats, Matrix (>= 1.2), dplyr (>= 0.7.8), plotly

Suggests knitr, rmarkdown, kableExtra, psych, psychTools, ggpubr, ggplot2, tidyselect, mvtnorm (>= 1.0.8), corpcor (>= 1.6.9), plyr (>= 1.8.4), matrixcalc (>= 1.0-3), pbapply, OpenMx, wTO, MASS, fdrtool, fitdistrplus

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 2020-07-13 04:40:02 UTC

R topics documented:

EGAnet-package	3
boot.wmt	3
bootEGA	4
CFA	7
cmi	8
depression	10
dimStability	10
dnn.weights	12
dynamic.plot	12
dynEGA	13
EBICglasso.qgraph	16
EGA	18
EGA.estimate	20
EGA.fit	23
ega.wmt	24
Embed	25
entropyFit	26
glla	27
intelligenceBattery	29
itemStability	29
LCT	31
net.loads	32
net.scores	34
node.redundant	36
node.redundant.combine	37
node.redundant.names	39
optimism	40
plots	41
prints	42
residualEGA	43
sim.dynEGA	44
simDFM	45
summaries	46
tefi	47
toy.example	48
vn.entropy	49
wmt2	50

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> and Alexander Christensen <alexpaulchristensen@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](https://doi.org/10.1371/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](https://doi.org/10.1016/j.intell.2017.02.007)
- Golino, H., Shi, D., Garrido, L. E., Christensen, A. P., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (in press). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*. doi: [10.31234/osf.io/gzcre](https://doi.org/10.31234/osf.io/gzcre)

boot.wmt

[bootEGA Results of wmt2Data](#)**Description**

[bootEGA](#) results using the "gLasso" model and 500 iterations of the Wiener Matrizen-Test 2 (WMT-2)

[bootEGA](#) Results of [wmt2Data](#)

Usage

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

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

Format

A list with 8 objects (see [bootEGA](#))

Details

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

Examples

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

```
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"),
  algorithm = c("walktrap", "louvain"),
  type = c("parametric", "resampling"),
  typicalStructure = TRUE,
  plot.typicalStructure = TRUE,
  ncores,
  ...
)
```

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:

	<ul style="list-style-type: none"> • "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
algorithm	A string indicating the algorithm to use. Current options are: <ul style="list-style-type: none"> • walktrap Computes the Walktrap algorithm using cluster_walktrap • louvain Computes the Walktrap algorithm using cluster_louvain
type	Character. A string indicating the type of bootstrap to use. Current options are: <ul style="list-style-type: none"> • "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 <i>n</i> 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 <i>n</i> bootstraps, and its estimated dimensions. Defaults to TRUE
ncores	Numeric. Number of cores to use in computing results. Defaults to <code>parallel::detectCores() / 2</code> or half of your computer's processing power. 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: <code>parallel::detectCores()</code>
...	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](https://doi.org/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

<code>ega.obj</code>	An EGA object
<code>data</code>	A dataframe with the variables to be used in the analysis
<code>estimator</code>	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
<code>plot.CFA</code>	Logical. Should the CFA structure with its standardized loadings be plot? Defaults to TRUE
<code>layout</code>	Layout of plot (see semPaths). Defaults to "spring"
<code>...</code>	Arguments passed to cfa

Value

Returns a list containing:

<code>fit</code>	Output from cfa
<code>summary</code>	Summary output from lavaan-class
<code>fit.measures</code>	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 <code>cluster_walktrap</code> 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](https://doi.org/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.

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

Usage

```
data(depression)
```

```
data(depression)
```

Format

A 574x78 response matrix

Examples

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

```
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 times the original dimension is exactly replicated in across bootstrap samples

Usage

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

Arguments

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

Value

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

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

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)

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

## End(Not run)
```

dnn.weights

Loadings Comparison Test Deep Learning Neural Network Weights

Description

A list of weights from four different neural network models: random vs. non-random model (`r_nr_weights`), low correlation factor vs. network model (`lf_n_weights`), high correlation with fewer variables than factors vs. network model (`hlf_n_weights`), and high correlation with greater variables than factors vs. network model (`hgf_n_weights`)

Usage

```
data(dnn.weights)
```

Format

A list of with a length of 4

Examples

```
data("dnn.weights")
```

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

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

Author(s)

Hudson F. Golino <[hfg9s at virginia.edu](mailto:hfg9s@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)
```

dynEGA

Dynamic Exploratory Graph Analysis

Description

Estimates dynamic factors in multivariate time series (i.e. longitudinal data, panel data, intensive longitudinal data) at multiple time scales, in different levels of analysis: individuals (intraindividual structure), groups or population (structure of the population). Exploratory graph analysis is applied in the derivatives estimated using generalized local linear approximation ([glla](#)). Instead of estimating factors by modeling how variables are covarying, as in traditional EGA, dynEGA is a dynamic model that estimates the factor structure by modeling how variables are changing together. GLLA is a filtering method for estimating derivatives from data that uses time delay embedding and a variant of Savitzky-Golay filtering to accomplish the task.

Usage

```
dynEGA(  
  data,  
  n.embed,  
  tau = 1,
```

```

delta = 1,
level = c("individual", "group", "population"),
id = NULL,
group = NULL,
use.derivatives = 1,
model = c("glasso", "TMFG"),
algorithm = c("walktrap", "louvain"),
plot.EGA = TRUE,
cor = c("cor_auto", "pearson", "spearman"),
steps = 4,
ncores
)

```

Arguments

<code>data</code>	A dataframe with the variables to be used in the analysis. The dataframe should be in a long format (i.e. observations for the same individual (for example, individual 1) are placed in order, from time 1 to time t, followed by the observations from individual 2, also ordered from time 1 to time t.)
<code>n.embed</code>	Integer. Number of embedded dimensions (the number of observations to be used in the Embed function). For example, an "n.embed = 5" will use five consecutive observations to estimate a single derivative.
<code>tau</code>	Integer. Number of observations to offset successive embeddings in the Embed function. A tau of one uses adjacent observations. Default is "tau = 1".
<code>delta</code>	Integer. The time between successive observations in the time series. Default is "delta = 1".
<code>level</code>	Character. A string indicating the level of analysis. If the interest is in modeling the intraindividual structure only (one dimensionality structure per individual), then <code>level</code> should be set to "individual". If the interest is in the structure of a group of individuals, then <code>level</code> should be set to "group". Finally, if the interest is in the population structure, then <code>level</code> should be set to "population". Current options are: <ul style="list-style-type: none"> • <code>individual</code> Estimates the dynamic factors per individual. This should be the preferred method if one is interested in the factor structure of individuals. An additional parameter ("<code>id</code>") needs to be provided identifying each individual. • <code>group</code> Estimates the dynamic factors for each group. An additional parameter ("<code>group</code>") needs to be provided identifying the group membership. • <code>population</code> Estimates the dynamic factors of the population
<code>id</code>	Numeric. Number of the column identifying each individual.
<code>group</code>	Numeric or character. Number of the column identifying group membership. Must be specified only if <code>level = "group"</code> .
<code>use.derivatives</code>	Integer. The order of the derivative to be used in the EGA procedure. Default to 1.

model	A string indicating the network method to use (EGA.estimate). Current options are: <ul style="list-style-type: none"> • <code>glasso</code> Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter. This is the default method • <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph
algorithm	A string indicating the community detection algorithm to use. Current options are: <ul style="list-style-type: none"> • <code>walktrap</code> Computes the Walktrap algorithm using cluster_walktrap • <code>louvain</code> Computes the Walktrap algorithm using cluster_louvain
plot.EGA	Logical. If TRUE, returns a plot of the network and its estimated dimensions. Defaults to TRUE
cor	Type of correlation matrix to compute. The default uses cor_auto . Current options are: <ul style="list-style-type: none"> • <code>cor_auto</code> Computes the correlation matrix using the cor_auto function from qgraph. • <code>pearson</code> Computes Pearson's correlation coefficient using the pairwise complete observations via the cor function. • <code>spearman</code> Computes Spearman's correlation coefficient using the pairwise complete observations via the cor function.
steps	Number of steps to be used in cluster_walktrap algorithm. Defaults to 4.
ncores	Numeric. Number of cores to use in computing results. Defaults to <code>parallel::detectCores() / 2</code> or half of your computer's processing power. 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: <code>parallel::detectCores()</code>

Author(s)

Hudson Golino <hfg9s at virginia.edu>

References

- Boker, S. M., Deboeck, P. R., Edler, C., & Keel, P. K. (2010) Generalized local linear approximation of derivatives from time series. In S.-M. Chow, E. Ferrer, & F. Hsieh (Eds.), *The Notre Dame series on quantitative methodology. Statistical methods for modeling human dynamics: An interdisciplinary dialogue*, (p. 161-178). *Routledge/Taylor & Francis Group*. doi:[10.1037/a0016622](#)
- Deboeck, P. R., Montpetit, M. A., Bergeman, C. S., & Boker, S. M. (2009) Using derivative estimates to describe intraindividual variability at multiple time scales. *Psychological Methods*, *14*(4), 367-386. doi:[10.1037/a0016622](#)
- 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](#)
- Savitzky, A., & Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, *36*(8), 1627-1639. doi:[10.1021/ac60214a047](#)

Examples

```
# Population structure:
dyn.random <- dynEGA(data = sim.dynEGA, n.embed = 5, tau = 1,
  delta = 1, id = 21, group = 22, use.derivatives = 1,
  level = "population", model = "glasso", ncores = 2)

# Group structure:
dyn.group <- dynEGA(data = sim.dynEGA, n.embed = 5, tau = 1,
  delta = 1, id = 21, group = 22, use.derivatives = 1,
  level = "group", model = "glasso", ncores = 2)

# Intraindividual structure:

dyn.individual <- dynEGA(data = sim.dynEGA, n.embed = 5, tau = 1,
  delta = 1, id = 21, group = 22, use.derivatives = 1,
  level = "individual", model = "glasso", ncores = 2)
```

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

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

Applies 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"),
  algorithm = c("walktrap", "louvain"),
  plot.EGA = TRUE,
  n = NULL,
  steps = 4,
  nvar = 4,
  nfact = 1,
  load = 0.7,
  ...
)
```

Arguments

<code>data</code>	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 <code>n</code> will need to be specified.
<code>model</code>	A string indicating the method to use. Current options are: <ul style="list-style-type: none"> • <code>glasso</code> Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter. This is the default method • <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph
<code>algorithm</code>	A string indicating the algorithm to use. Current options are: <ul style="list-style-type: none"> • <code>walktrap</code> Computes the Walktrap algorithm using <code>cluster_walktrap</code> • <code>louvain</code> Computes the Walktrap algorithm using <code>cluster_louvain</code>
<code>plot.EGA</code>	Logical. If TRUE, returns a plot of the network and its estimated dimensions. Defaults to TRUE
<code>n</code>	Integer. Sample size, if the data provided is a correlation matrix
<code>steps</code>	Number of steps to be used in <code>cluster_walktrap</code> algorithm. Defaults to 4.
<code>nvar</code>	Number of variables to use in the simulation part of the unidimensionality check. Defaults to 4.
<code>nfact</code>	Number of factors to be simulated (part of the unidimensionality check algorithm). Defaults to 1.
<code>load</code>	Factor loadings (used in the unidimensionality check algorithm). Defaults to 0.70.
<code>...</code>	Additional arguments to be passed to <code>EBICglasso.qgraph</code> or <code>TMFG</code>

Details

This algorithm includes checking for whether the data is unidimensional.

Value

Returns a list containing:

<code>network</code>	A symmetric network estimated using either the <code>EBICglasso.qgraph</code> or <code>TMFG</code>
<code>wc</code>	A vector representing the community (dimension) membership of each node in the network. NA values mean that the node was disconnected from the network
<code>n.dim</code>	A scalar of how many total dimensions were identified in the network
<code>cor.data</code>	The zero-order correlation matrix

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](https://doi.org/10.1371/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](https://doi.org/10.1016/j.intell.2017.02.007)
- Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (in press). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*. doi: [10.31234/osf.io/gzcre](https://doi.org/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)
```

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.estimate(
  data,
  n = NULL,
  model = c("glasso", "TMFG"),
  algorithm = c("walktrap", "louvain"),
  steps = 4,
  cor = c("cor_auto", "pearson", "spearman"),
  ...
)
```

Arguments

<code>data</code>	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 <code>n</code> will need to be specified.
<code>n</code>	Integer. Sample size, if the data provided is a correlation matrix
<code>model</code>	A string indicating the method to use. Current options are: <ul style="list-style-type: none"> • <code>glasso</code> Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter. This is the default method • <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph
<code>algorithm</code>	A string indicating the algorithm to use. Current options are: <ul style="list-style-type: none"> • <code>walktrap</code> Computes the Walktrap algorithm using cluster_walktrap • <code>louvain</code> Computes the Walktrap algorithm using cluster_louvain
<code>steps</code>	Number of steps to be used in cluster_walktrap algorithm. Defaults to 4.
<code>cor</code>	Type of correlation matrix to compute. The default uses <code>cor_auto</code> . Current options are: <ul style="list-style-type: none"> • <code>cor_auto</code> Computes the correlation matrix using the <code>cor_auto</code> function from qgraph. • <code>pearson</code> Computes Pearson's correlation coefficient using the pairwise complete observations via the <code>cor</code> function. • <code>spearman</code> Computes Spearman's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.
<code>...</code>	Additional arguments to be passed to EBICglasso.qgraph or <code>TMFG</code>

Value

Returns a list containing:

estimated.network	A symmetric network estimated using either the EBICglasso.qgraph or TMFG
wc	A vector representing the community (dimension) membership of each node in the network. NA values mean that the node was disconnected from the network
n.dim	A scalar of how many total dimensions were identified in the network
cor.data	The zero-order correlation matrix

Author(s)

Alexander P. Christensen <alexpaulchristensen@gmail.com> and Hudson F. 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](https://doi.org/10.1371/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](https://doi.org/10.1016/j.intell.2017.02.007)
- Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (in press). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*. doi: [10.31234/osf.io/gzcre](https://doi.org/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.estimate(data = wmt2[,7:24], model = "glasso")

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

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

EGA.fit [EGA Optimal Model Fit using the Total Entropy Fit Index \(tefi\)](#)

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](#). Also computes [cluster_louvain](#) community detection algorithm.

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: <ul style="list-style-type: none"> • "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](https://doi.org/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 WMT-2 Data

Description

[EGA](#) Network of [wmt2Data](#)

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

Usage

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

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


Format

A 17 x 17 adjacency matrix

Details

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

Examples

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

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

Embed

Time-delay Embedding

Description

Reorganizes an individual's observed time series into an embedded matrix. The embedded matrix is constructed with replicates of an individual time series that are offset from each other in time. The function requires two parameters, one that specifies the number of observations to be used (i.e. the number of embedded dimensions) and the other that specifies the number of observations to offset successive embeddings.

Usage

```
Embed(x, E, tau)
```

Arguments

x	Vector. An observed time series to be reorganized into a time-delayed embedded matrix.
E	Integer. Number of embedded dimensions or the number of observations to be used. For example, an "E = 5" will generate a matrix with five columns, meaning that five consecutive observations are used to create each row of the embedded matrix.
tau	Integer. Number of observations to offset successive embeddings. A tau of one uses adjacent observations. Default is "tau = 1".

Value

Returns a matrix containing the embedded matrix.

Author(s)

Pascal Deboeck <pascal.deboeck@psych.utah.edu>

References

Deboeck, P. R., Montpetit, M. A., Bergeman, C. S., & Boker, S. M. (2009) Using derivative estimates to describe intraindividual variability at multiple time scales. *Psychological Methods*, *14*(4), 367-386. doi:[10.1037/a0016622](https://doi.org/10.1037/a0016622)

Examples

```
# A time series with 8 time points
tseries <- 49:56
embed.tseries <- Embed(tseries, E = 4, tau = 1)
```

entropyFit	<i>Entropy Fit Index</i>
------------	--------------------------

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)
```

glla

Generalized Local Linear Approximation

Description

Estimates the derivatives of a time series using generalized local linear approximation (GLLA). GLLA is a filtering method for estimating derivatives from data that uses time delay embedding and a variant of Savitzky-Golay filtering to accomplish the task.

Usage

```
glla(x, n.embed, tau, delta, order)
```

Arguments

x	Vector. An observed time series.
n.embed	Integer. Number of embedded dimensions (the number of observations to be used in the Embed function).
tau	Integer. Number of observations to offset successive embeddings in the Embed function. A tau of one uses adjacent observations. Default is "tau = 1".
delta	Integer. The time between successive observations in the time series. Default is "delta = 1".
order	Integer. The maximum order of the derivative to be estimated. For example, "order = 2" will return a matrix with three columns with the estimates of the observed scores and the first and second derivative for each row of the embedded matrix (i.e. the reorganization of the time series implemented via the Embed function).

Value

Returns a matrix containing n columns, in which n is one plus the maximum order of the derivatives to be estimated via generalized local linear approximation.

Author(s)

Hudson Golino <hfg9s at virginia.edu>

References

- Boker, S. M., Deboeck, P. R., Edler, C., & Keel, P. K. (2010) Generalized local linear approximation of derivatives from time series. In S.-M. Chow, E. Ferrer, & F. Hsieh (Eds.), *The Notre Dame series on quantitative methodology. Statistical methods for modeling human dynamics: An interdisciplinary dialogue*, (p. 161–178). *Routledge/Taylor & Francis Group*. doi:[10.1037/a0016622](https://doi.org/10.1037/a0016622)
- Deboeck, P. R., Montpetit, M. A., Bergeman, C. S., & Boker, S. M. (2009) Using derivative estimates to describe intraindividual variability at multiple time scales. *Psychological Methods*, *14*(4), 367-386. doi:[10.1037/a0016622](https://doi.org/10.1037/a0016622)
- Savitzky, A., & Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, *36*(8), 1627-1639. doi:[10.1021/ac60214a047](https://doi.org/10.1021/ac60214a047)

Examples

```
# A time series with 8 time points
tseries <- 49:56
deriv.tseries <- glla(tseries, n.embed = 4, tau = 1, delta = 1, order = 2)
```

intelligenceBattery *Intelligence Data*

Description

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

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

Usage

```
data(intelligenceBattery)
```

```
data(intelligenceBattery)
```

Format

A 1185x125 response matrix

Examples

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

```
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
mean.dim.rep	The average replication proportion of items replicating in each dimension. More simply, the average of the item.replication output for each 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)

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

## End(Not run)

```

LCT

Loadings Comparison Test

Description

An algorithm to identify whether data were generated from a random, factor, or network model using factor and network loadings. The algorithm uses heuristics based on theory and simulation. These heuristics were then submitted to several deep learning neural networks with 240,000 samples per model with varying parameters.

Usage

```
LCT(data, n, iter = 100)
```

Arguments

<code>data</code>	Matrix or data frame. A dataframe with the variables to be used in the test or a correlation matrix. If the data used is a correlation matrix, the argument <code>n</code> will need to be specified
<code>n</code>	Integer. Sample size (if the data provided is a correlation matrix)
<code>iter</code>	Integer. Number of replicate samples to be drawn from a multivariate normal distribution (uses <code>mvtnorm::mvtnorm</code>). Defaults to 100

Value

Returns a list containing:

<code>empirical</code>	Prediction of model based on empirical dataset only
------------------------	---

bootstrap	Prediction of model based on means of the loadings across the bootstrap replicate samples
proportions	Proportions of models suggested across bootstraps
omnibus	An omnibus prediction based on a consensus of empirical, bootstrap, and bootstrap proportions prediction. A consensus corresponds to any combination of two predictions returning the same prediction

Author(s)

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

References

Christensen, A. P., & Golino, H. (2020). On the equivalency of factor and network loadings. *PsyArXiv*. doi:[10.31234/osf.io/xakez](https://doi.org/10.31234/osf.io/xakez)

Examples

```
# Compute LCT
## Network model
LCT(data = wmt2[,7:24])$omnibus

## Factor model
LCT(data = NetworkToolbox::neoOpen)$omnibus
```

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, pos.manifold = FALSE, min.load = 0, plot = FALSE)
```

Arguments

A	Matrix, data frame, or EGA object. An adjacency matrix of network data
wc	Numeric or character vector. A vector of community assignments. If input into A is an EGA object, then wc is automatically detected
pos.manifold	Boolean. Should a positive manifold be applied (i.e., should all dimensions be positively correlated)? Defaults to FALSE. Set to TRUE for a positive manifold

min.load	Numeric. Sets the minimum loading allowed in the standardized network loading matrix. Values equal or greater than the minimum loading are kept in the output. Values less than the minimum loading are removed. This matrix can be viewed using <code>print()</code> or <code>summary()</code> . Defaults to 0
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
minLoad	The minimum loading to appear in summary of network loadings. Use <code>print()</code> or <code>summary()</code> to view
plot	A <code>qgraph</code> plot of the network loadings. Use <code>plot</code> to view

Author(s)

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

References

- Christensen, A. P., & Golino, H. (2020). On the equivalency of factor and network loadings. *PsyArXiv*. doi:[10.31234/osf.io/xakez](https://doi.org/10.31234/osf.io/xakez)
- Christensen, A. P., Golino, H., & Silvia, P. (2019). A psychometric network perspective on the measurement and assessment of personality traits. *PsyArXiv*. doi:[10.31234/osf.io/ktejp](https://doi.org/10.31234/osf.io/ktejp)
- Hallquist, M., Wright, A. C. G., & Molenaar, P. C. M. (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](https://doi.org/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)
```

net.scores

Network Scores

Description

This function computes network scores computed based on each node's [strength](#) within each community (i.e., factor) in the network (see [net.loads](#)). These values are used as network "factor loadings" for the weights of each item. Notably, network analysis allows nodes to contribute to more than one community. These loadings are considered in the network 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 `global`). An important difference is that the network scores account for cross-loadings in their estimation of scores

Usage

```
net.scores(data, A, wc, global = FALSE, impute = "none", ...)
```

Arguments

<code>data</code>	Matrix or data frame. Must be a dataset
<code>A</code>	Matrix, data frame, or EGA object. An adjacency matrix of network data
<code>wc</code>	Numeric. A vector of community assignments. Not necessary if an EGA object is input for argument <code>A</code>
<code>global</code>	Boolean. Should general network loadings be computed in scores? Defaults to FALSE. 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
<code>impute</code>	Character. In the presence of missing data, imputation can be implemented. Currently, three options are available: <ul style="list-style-type: none"> • none No imputation is performed. This is the default.

- mean The "mean" value of the columns are used to replace the missing data.
- median The "median" value of the columns are used to replace the missing data.

... Additional arguments for [EGA](#)

Details

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

Value

Returns a list containing:

<code>unstd.scores</code>	The unstandardized network scores for each participant and community (including the overall score)
<code>std.scores</code>	The standardized network scores for each participant and community (including the overall score)
<code>commCor</code>	Partial correlations between the specified or identified communities
<code>loads</code>	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@virginia.edu>

References

Christensen, A. P., & Golino, H. (under review). On the equivalency of factor and network loadings. *PsyArXiv*. doi: [10.31234/osf.io/xakez](https://doi.org/10.31234/osf.io/xakez)

Christensen, A. P., Golino, H. F., & Silvia, P. J. (in press). A psychometric network perspective on the measurement and assessment of personality traits. *European Journal of Personality*. doi: [10.1002/per.2265](https://doi.org/10.1002/per.2265)

Golino, H., Christensen, A. P., Moulder, R., Kim, S., & Boker, S. M. (under review). Modeling latent topics in social media using Dynamic Exploratory Graph Analysis: The case of the right-wing and left-wing trolls in the 2016 US elections. *PsyArXiv*. doi: [10.31234/osf.io/tfs7c](https://doi.org/10.31234/osf.io/tfs7c)

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	<i>Detects Redundant Nodes in a Network</i>
----------------	---

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,
  n = NULL,
  sig,
  method = c("wTO", "pcor", "thresh"),
  type = c("alpha", "bonferroni", "FDR", "adapt")
)
```

Arguments

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

Value

Returns a list:

<code>redundant</code>	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

simulation using node.redundant Christensen, A. P. (2020). Towards a network psychometrics approach to assessment: Simulations for redundancy, dimensionality, and loadings (Unpublished doctoral dissertation). University of North Carolina at Greensboro, Greensboro, NC, USA. <https://doi.org/10.31234/osf.io/84kgd>

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. <https://doi.org/10.1073/pnas.0911376106>

Examples

```
# obtain SAPA items
items <- psychTools::spi[,c(11:20)]

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

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

# threshold
redund <- node.redundant(items, method = "thresh", sig = .20)
```

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", "latent"),
  estimator = "WLSMV",
  auto = FALSE,
  ...
)
```

Arguments

node.redundant.obj	A node.redundant object
type	Character. Method to use to combine redundant variables. <ul style="list-style-type: none"> • "sum" Computes sum scores (i.e., means) of the variables • "latent" Computes latent variable scores using <code>[lavaan]{cfa}</code> Defaults to "latent"
estimator	Character. Estimator to use for latent variables. Defaults to "WLSMV". See <code>[lavaan]{cfa}</code> 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: <ol style="list-style-type: none"> 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 <code>[lavaan]{cfa}</code>

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, method = "wTO", type = "adapt")

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

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

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)
```

```
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](https://doi.org/10.1037//0022-3514.67.6.1063)

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](https://doi.org/10.1037//0022-3514.67.6.1063)

Examples

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

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

plots

S3Methods for Plotting

Description

Plots for EGAnet objects

Usage

```
## S3 method for class 'bootEGA'  
plot(x, vsize = 6, ...)  
  
## S3 method for class 'CFA'  
plot(x, layout = "spring", vsize = 6, ...)  
  
## S3 method for class 'dynEGA'  
plot(x, title = "", vsize = 6, ...)  
  
## S3 method for class 'dynEGA.Groups'  
plot(x, ncol, nrow, title = "", vsize = 6, ...)  
  
## S3 method for class 'dynEGA.Individuals'  
plot(x, title = "", vsize = 6, id = NULL, ...)  
  
## S3 method for class 'EGA'  
plot(x, title = "", vsize = 6, ...)  
  
## S3 method for class 'NetLoads'  
plot(x, ...)
```

Arguments

x	Object from EGAnet package
vsize	Numeric. Size of vertices in network plots. Defaults to 6
...	Arguments passed on to <ul style="list-style-type: none"> • qgraph Functions: bootEGA, dynEGA, dynEGA.Groups, dynEGA.Individuals, EGA, and net.loads • semPaths Functions: CFA
layout	Character. Layout of plot (see semPaths). Defaults to "spring"
ncol	Numeric. Number of columns
nrow	Numeric. Number of rows
title	Character. Title of the plot. Defaults to ""
id	Numeric. An integer or character indicating the ID of the individual to plot

Value

Plots of EGAnet object

Author(s)

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

prints

S3Methods for Printing

Description

Prints for EGAnet objects

Usage

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

## S3 method for class 'dynEGA.Groups'
print(x, ...)

## S3 method for class 'dynEGA.Individuals'
print(x, ...)

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

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

Arguments

x Object from EGAnet package
 ... Additional arguments

Value

Prints EGAnet object

Author(s)

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

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 <code>ggmfit</code> 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)
```

sim.dynEGA

sim.dynEGA Data

Description

A simulated (multivariate time series) data with 20 variables, 200 individual observations, 50 time points per individual and 2 groups of individuals.

Usage

```
data(sim.dynEGA)
```

Format

A 10000x22 multivariate time series

Examples

```
data("sim.dynEGA")
```

simDFM

*Simulate data following a Dynamic Factor Model***Description**

Function to simulate data following a dynamic factor model (DFM). Two DFMs are currently available: the direct autoregressive factor score model (Engle & Watson, 1981; Nesselroade, McArdle, Aggen, and Meyers, 2002) and the dynamic factor model with random walk factor scores.

Usage

```
simDFM(
  variab,
  timep,
  nfact,
  error,
  dfm = c("DAFS", "RandomWalk"),
  loadings,
  autoreg,
  crossreg,
  var.shock,
  cov.shock,
  burnin = 1000
)
```

Arguments

variab	Number of variables per factor.
timep	Number of time points.
nfact	Number of factors.
error	Value to be used to construct a diagonal matrix Q. This matrix is p x p covariance matrix Q that will generate random errors following a multivariate normal distribution with mean zeros. The value provided is squared before constructing Q.
dfm	A string indicating the dynamical factor model to use. Current options are: <ul style="list-style-type: none"> • DAFS Simulates data using the direct autoregressive factor score model. This is the default method • RandomWalk Simulates data using a dynamic factor model with random walk factor scores.
loadings	Magnitude of the loadings.
autoreg	Magnitude of the autoregression coefficients.
crossreg	Magnitude of the cross-regression coefficients.
var.shock	Magnitude of the random shock variance.
cov.shock	Magnitude of the random shock covariance
burnin	Number of n first samples to discard when computing the factor scores. Defaults to 1000.

Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

References

A one-factor multivariate time series model of metropolitan wage rates. *Journal of the American Statistical Association*, 76(376), 774-781. ref:01621459.1981.10477720#.XmkWWy2ZMuA

Nesselroade, J. R., McArdle, J. J., Aggen, S. H., & Meyers, J. M. (2002). Dynamic factor analysis models for representing process in multivariate time-series. In D. S. Moskowitz & S. L. Hershberger (Eds.), *Multivariate applications book series. Modeling intraindividual variability with repeated measures data: Methods and applications*, 235–265. <https://psycnet.apa.org/record/2001-05300-009>

Examples

```
## Not run:
\donttest{
# Estimate EGA network
data1 <- simDFM(variab = 5, timep = 50, nfact = 3, error = 0.05,
dfm = "DAFS", loadings = 0.7, autoreg = 0.8,
crossreg = 0.1, var.shock = 0.18,
cov.shock = 0.36, burnin = 1000)
}

## End(Not run)
```

summarys

S3Methods for Summaries

Description

Summaries for EGAnet objects

Usage

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

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

## S3 method for class 'dynEGA.Groups'
summary(object, ...)

## S3 method for class 'dynEGA.Individuals'
```

```
summary(object, ...)

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

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

Arguments

```
object      Object from EGAnet package
...         Additional arguments
```

Value

Summarizes EGAnet object

Author(s)

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

tefi	<i>Total Entropy Fit Index using Von Neumman's entropy (Quantum Information Theory) for correlation matrices</i>
------	--

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 dataframe or 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 = ega.wmt$correlation, structure = ega.wmt$wc)
```

toy.example

Toy Example Data

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	<i>Entropy Fit Index using Von Neumman's entropy (Quantum Information Theory) for correlation matrices</i>
------------	--

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 dataframe 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@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 = ega.wmt$correlation, structure = ega.wmt$wc)
```

wmt2

WMT-2 Data

Description

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

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

Usage

```
data(wmt2)
```

```
data(wmt2)
```

Format

A 1185x24 response matrix

Examples

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

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

Index

* datasets

- boot.wmt, 3
 - depression, 10
 - dnn.weights, 12
 - ega.wmt, 24
 - intelligenceBattery, 29
 - optimism, 40
 - sim.dynEGA, 44
 - toy.example, 48
 - wmt2, 50
- adapt.a, 36
- boot.wmt, 3
- bootEGA, 3, 4, 4, 7, 9–11, 20, 22, 24, 29, 30
- CFA, 6, 7, 11, 13, 20, 22, 24, 27, 30, 44, 48, 49
- cfa, 7
- cluster_louvain, 5, 15, 19, 21, 23
- cluster_walktrap, 5, 9, 15, 18, 19, 21, 23
- cmi, 8
- comcat, 32
- cor, 15, 21
- cor_auto, 8, 15, 21
- depression, 10
- dimStability, 10
- dnn.weights, 12
- dynamic.plot, 12
- dynEGA, 13
- EBICglasso, 16, 36, 37
- EBICglasso.qgraph, 5, 16, 18, 19, 21–23
- EGA, 4–7, 9, 11–13, 18, 23–27, 29, 30, 32, 34, 35, 43, 44, 47–49
- EGA.estimate, 15, 20
- EGA.fit, 23
- ega.wmt, 24
- EGAnet (EGAnet-package), 3
- EGAnet-package, 3
- Embed, 14, 25, 28
- entropyFit, 26
- fdrtool, 36
- fitMeasures, 7
- glasso, 16, 17
- glla, 13, 27
- intelligenceBattery, 29
- itemStability, 29
- lavOptions, 7
- LCT, 31
- Mvnorm, 5
- net.loads, 30, 32, 34, 35
- net.scores, 34
- node.redundant, 36, 38, 39
- node.redundant.combine, 37
- node.redundant.names, 39
- optimism, 40
- plot.bootEGA (plots), 41
- plot.CFA (plots), 41
- plot.dynEGA (plots), 41
- plot.EGA (plots), 41
- plot.NetLoads (plots), 41
- plotly, 12
- plots, 41
- print.dynEGA (prints), 42
- print.EGA (prints), 42
- print.NetLoads (prints), 42
- prints, 42
- qgraph, 8, 15, 16, 21, 33, 42, 44
- residualEGA, 43
- semPaths, 7, 42
- sim.dynEGA, 44

simDFM, [45](#)
stable, [32](#)
strength, [32](#), [34](#)
summary.CFA (summary), [46](#)
summary.dynEGA (summary), [46](#)
summary.EGA (summary), [46](#)
summary.NetLoads (summary), [46](#)
summary, [46](#)

tefi, [23](#), [47](#)
TMFG, [5](#), [18](#), [19](#), [21–23](#)
toy.example, [48](#)

vn.entropy, [49](#)

wi2net, [17](#)
wmt2, [3](#), [24](#), [50](#)
wTO, [36](#)