

Package ‘EGAnet’

October 12, 2022

Title Exploratory Graph Analysis – a Framework for Estimating the Number of Dimensions in Multivariate Data using Network Psychometrics

Version 1.2.3

Date 2022-09-05

Maintainer Hudson Golino <hfg9s@virginia.edu>

Description Implements the Exploratory Graph Analysis (EGA) framework for dimensionality and psychometric assessment. EGA is part of a new area called network psychometrics that uses undirected network models for the assessment of psychometric properties. EGA estimates the number of dimensions (or factors) using graphical lasso or Triangulated Maximally Filtered Graph (TMFG) and a weighted network community detection algorithm. A bootstrap method for verifying the stability of the dimensions and items in those dimensions is available. The fit of the structure suggested by EGA can be verified using Entropy Fit Indices. A novel approach called Unique Variable Analysis (UVA) can be used to identify and reduce redundant variables in multivariate data. Network loadings, which are roughly equivalent to factor loadings when the data generating model is a factor model, are available. Network scores can also be computed using the network loadings. Dynamic EGA (dynEGA) will estimate dimensions from time series data for individual, group, and sample levels. Golino, H., & Epskamp, S. (2017) <doi:10.1371/journal.pone.0174035>. Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (2020) <doi:10.31234/osf.io/gzcre>. Christensen, A. P., & Golino, H. (under review) <doi:10.31234/osf.io/hz89e>. Golino, H., Moulder, R. G., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Nesselroade, J., Sadana, R., Thiyagarajan, J. A., & Boker, S. M. (2020) <doi:10.31234/osf.io/mtka2>. Christensen, A. P. & Golino, H. (2021) <doi:10.3390/psych3030032>. Christensen, A. P., Garrido, L. E., & Golino, H. (under review) <doi:10.31234/osf.io/4kra2>. Golino, H., Christensen, A. P., Moulder, R. G., Kim, S., & Boker, S. M. (under review) <doi:10.31234/osf.io/tfs7c>.

Depends R (>= 3.5.0)

License GPL (>= 3.0)

Encoding UTF-8

LazyData true

Imports glasso, GGally, gg dendro, ggplot2, ggpubr, igraph (>= 1.3.0), lavaan, Matrix, matrixcalc, methods, network, OpenMx, pbapply, qgraph, semPlot, stats

Suggests GPARotation, gridExtra, knitr, markdown, psych, psychTools,
pwr, RColorBrewer, rmarkdown, rstudioapi, sna

VignetteBuilder knitr

RoxygenNote 7.2.1

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>>),
Luis E. Garrido [ctb] (<<https://orcid.org/0000-0001-8932-6063>>),
Laura Jamison [ctb] (<<https://orcid.org/0000-0002-4656-8684>>)

Repository CRAN

Date/Publication 2022-09-05 15:50:02 UTC

R topics documented:

EGAnet-package	3
boot.ergoInfo	5
boot.wmt	7
bootEGA	7
CFA	12
color_palette_EGA	14
compare.EGA.plots	15
convert2igraph	17
depression	17
dimensionStability	18
dnn.weights	19
dynEGA	20
dynEGA.ind.pop	23
EBICglasso.qgraph	26
EGA	28
EGA.estimate	32
EGA.fit	36
ega.wmt	38
Embed	39
entropyFit	40
ergoInfo	41
glla	43
hierEGA	44
infoCluster	48
intelligenceBattery	49
invariance	50
itemStability	53
jsd	55
jsd.ergoInfo	56
LCT	58
louvain	59

mctest.ergoInfo	61
methods.section	64
net.loads	66
net.scores	68
network.descriptives	70
optimism	71
plots	72
prime.num	73
prints	74
residualEGA	75
riEGA	76
sim.dynEGA	79
simDFM	80
summarys	81
tefi	82
TMFG	84
totalCor	85
totalCorMat	86
toy.example	87
UVA	87
vn.entropy	91
wmt2	93
Index	94

 EGAnet-package

 EGAnet-package

Description

Implements the Exploratory Graph Analysis (EGA; Golino & Epskamp, 2017; Golino, Shi et al., 2020) framework for dimensionality and psychometric assessment. EGA is part of a new area called *network psychometrics* that uses undirected network models for the assessment of psychometric properties. EGA estimates the number of dimensions (or factors) using graphical lasso [EBICglasso](#) or Triangulated Maximally Filtered Graph (TMFG) and a weighted network community detection algorithm (Christensen, Garrido, Golino, under review A). A bootstrap method for verifying the stability of the dimensions and items in those dimensions is available ([bootEGA](#); Christensen & Golino, 2021a). The fit of the structure suggested by EGA can be verified using Entropy Fit Indices ([entropyFit](#), [tefi](#); Golino, Moulder et al., 2020). A novel approach called Unique Variable Analysis (UVA) can be used to identify and reduce redundant variables in multivariate data (Christensen, Garrido, & Golino, under review B). Network loadings ([net.loads](#)), which are roughly equivalent to factor loadings when the data generating model is a factor model, are available (Christensen & Golino, 2021b, 2021c). Network scores ([net.scores](#)) can also be computed using the network loadings. Finally, dynamic EGA ([dynEGA](#)) will estimate dimensions from time series data for individual, group, and sample levels (Golino, Christensen et al., 2021).

Author(s)

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

References

Christensen, A. P., Garrido, L. E., & Golino, H. (under review A). Comparing community detection algorithms in psychological data: A Monte Carlo simulation. *PsyArXiv*.

Related functions: [EGA](#)

Christensen, A. P., Garrido, L. E., & Golino, H. (under review B). Unique Variable Analysis: A novel approach to detect redundant variables in multivariate data. *PsyArXiv*.

Related functions: [UVA](#)

Christensen, A. P., & Golino, H. (2021a). Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis: A tutorial. *Psych*, 3(3), 479-500.

Related functions: [bootEGA](#), [dimensionStability](#), # and [itemStability](#)

Christensen, A. P., & Golino, H. (2021b). Factor or network model? Predictions from neural networks. *Journal of Behavioral Data Science*, 1(1), 85-126.

Related functions: [LCT](#)

Christensen, A. P., & Golino, H. (2021c). On the equivalency of factor and network loadings. *Behavior Research Methods*, 53, 1563-1580.

Related functions: [LCT](#) and [net.loads](#)

Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, 34, 1095-1108.

Related functions: [bootEGA](#), [dimensionStability](#), # [EGA](#), [itemStability](#), and [UVA](#)

Golino, H., Christensen, A. P., Moulder, R., Kim, S., & Boker, S. M. (2021). 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. *Psychometrika*.

Related functions: [dynEGA](#) and [simDFM](#)

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

Related functions: [EGA](#)

Golino, H., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PLoS ONE*, 12, e0174035.

Related functions: [EGA](#)

Golino, H., Moulder, R., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Nesselroade, J., Sadana, R., Thiyagarajan, J. A., & Boker, S. M. (2020). Entropy fit indices: New fit measures for assessing the structure and dimensionality of multiple latent variables. *Multivariate Behavioral Research*.

Related functions: [entropyFit](#), [tefi](#), and [vn.entropy](#)

Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., Thiyagarajan, J. A., & Martinez-Molina, A. (2020). Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, 25, 292-320.

Related functions: [EGA](#)

Golino, H., Thiyagarajan, J. A., Sadana, M., Teles, M., Christensen, A. P., & Boker, S. M. (under review). Investigating the broad domains of intrinsic capacity, functional ability, and environment: An exploratory graph analysis approach for improving analytical methodologies for measuring healthy aging. *PsyArXiv*.

Related functions: [EGA.fit](#) and [tefi](#)

Jamison, L., Christensen, A. P., & Golino, H. (under review). Optimizing Walktrap's community detection in networks using the Total Entropy Fit Index. *PsyArXiv*.

Related functions: [EGA.fit](#) and [tefi](#)

boot.ergoInfo

Bootstrap Test for the Ergodicity Information Index

Description

Tests the Ergodicity Information Index obtained in the empirical sample with a distribution of EII obtained by bootstrap sampling. In traditional bootstrap sampling, individual participants are re-sampled with replacement from the empirical sample. This process is time consuming when carried out across v number of variables, n number of participants, t number of time points, and i number of iterations.

A more efficient process, the approach applied here, is to obtain a sampling distribution of EII values as if all participants in the data have the population network structure. Sampling is not perfect and therefore random noise is added to the edges of the population structure to simulate sampling variability. This noise follows a random uniform distribution ranging from -0.10 to 0.10. In addition, a proportion of edges are rewired to allow for slight variations on the population structure. The proportion of nodes that are rewired is sampled from a random uniform distribution between 0.20 to 0.40. This process is carried out for each participant resulting in n variations of the population structure. Afterward, EII is computed. This process is carried out for i iterations (e.g., 100).

The result is a sampling distribution of EII values that would be expected if the process was ergodic. If the empirical EII value is significantly less than the distribution or not significantly different, then the empirical data can be expected to be generated from an ergodic process and the population structure is sufficient to describe all individuals. If the empirical EII value is significantly greater than the distribution, then the empirical data cannot be described by the population structure – significant information is lost when collapsing across to the population structure.

Usage

```
boot.ergoInfo(dynEGA.object, EII, iter = 100, ncores, progress = TRUE)
```

Arguments

dynEGA.object	A dynEGA or a dynEGA.ind.pop object that is used to match the arguments of the EII object.
EII	A ergoInfo object, used to estimate the Empirical Ergodicity Information Index, or the estimated value of EII estimated using the ergoInfo function. Inherits use from ergoInfo

iter	Numeric integer. Number of replica samples to generate from the bootstrap analysis. At least 100 is recommended
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>
progress	Boolean. Should progress be displayed? Defaults to TRUE. For Windows, FALSE is about 2x faster

Value

Returns a list containing:

boot.ergoInfo	The values of the Ergodicity Information Index obtained in the bootstrap
p.value	The two-sided *p*-value of the bootstrap test for the Ergodicity Information Index. The null hypothesis is that the empirical Ergodicity Information index is equal to the expected value of the EII with small variation in the population structure
effect	Indicates whether the empirical EII is greater or less than the bootstrap distribution of EII.
interpretation	How you can interpret the result of the test in plain English
plot.dist	Histogram of the bootstrapped ergodicity information index
methods	Methods to report for print/summary S3methods and automated Methods section

Author(s)

Hudson Golino <hfg9s at virginia.edu> & Alexander P. Christensen <alexander.christensen at Vanderbilt.Edu>

References

Golino, H., Nesselrode, J., & Christensen, A. P. (2022). Toward a psychology of individuals: The ergodicity information index and a bottom-up approach for finding generalizations. *PsyArXiv*.

Examples

```
# Obtain simulated data
sim.data <- sim.dynEGA

## Not run:
# Dynamic EGA individual and population structures
dyn1 <- dynEGA.ind.pop(
  data = sim.dynEGA[,-26], n.embed = 5, tau = 1,
  delta = 1, id = 25, use.derivatives = 1,
  model = "glasso", ncores = 2, corr = "pearson"
)
```

```
# Empirical Ergodicity Information Index
eii1 <- ergoInfo(dynEGA.object = dyn1, use = "weighted")

# Bootstrap Test for Ergodicity Information Index
testing.ergoInfo <- boot.ergoInfo(
  dynEGA.object = dyn1, EII = eii1,
  ncores = 2
)
## End(Not run)
```

boot.wmt

[bootEGA Results of wmt2Data](#)

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 9 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 = NULL,
  uni.method = c("expand", "LE", "louvain"),
  iter,
  type = c("parametric", "resampling"),
  seed = 1234,
  corr = c("cor_auto", "pearson", "spearman"),
  EGA.type = c("EGA", "EGA.fit", "hierEGA", "riEGA"),
  model = c("glasso", "TMFG"),
  model.args = list(),
  algorithm = c("walktrap", "leiden", "louvain"),
  algorithm.args = list(),
  consensus.method = c("highest_modularity", "most_common", "iterative", "lowest_tefi"),
  consensus.iter = 100,
  typicalStructure = TRUE,
  plot.typicalStructure = TRUE,
  plot.args = list(),
  ncores,
  progress = TRUE,
  ...
)
```

Arguments

<code>data</code>	Matrix or data frame. Includes the variables to be used in the bootEGA analysis
<code>n</code>	Integer. Sample size if data provided is a correlation matrix
<code>uni.method</code>	Character. What unidimensionality method should be used? Defaults to "louvain". Current options are: <ul style="list-style-type: none"> • <code>expand</code> Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation. • <code>LE</code> Applies the Leading Eigenvalue algorithm (cluster_leading_eigen) on the empirical correlation matrix. If the number of dimensions is 1, then the Leading Eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation. • <code>louvain</code> Applies the Louvain algorithm (cluster_louvain) on the empirical correlation matrix using a resolution parameter = 0.95. If the number of dimensions is 1, then the Louvain solution is used; otherwise, regular EGA is used. This method was validated in the Christensen (2022) simulation.
<code>iter</code>	Numeric integer. Number of replica samples to generate from the bootstrap analysis. At least 500 is recommended

type	<p>Character. A string indicating the type of bootstrap to use.</p> <p>Current options are:</p> <ul style="list-style-type: none"> • "parametric" Generates n new datasets (multivariate normal random distributions) based on the original dataset, via the <code>mvrnorm</code> function • "resampling" Generates n random subsamples of the original data
seed	<p>Numeric. Seed to reproduce results. Defaults to 1234. For random results, set to NULL</p>
corr	<p>Character. Type of correlation matrix to compute. The default uses <code>cor_auto</code>.</p> <p>Current options are:</p> <ul style="list-style-type: none"> • <code>cor_auto</code> Computes the correlation matrix using the <code>cor_auto</code> function from <code>qgraph</code>. • <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.
EGA.type	<p>Character. Type of EGA model to use.</p> <p>Current options are:</p> <ul style="list-style-type: none"> • <code>EGA</code> Uses standard exploratory graph analysis • <code>EGA.fit</code> Uses <code>tefi</code> to determine best fit of <code>EGA</code> • <code>hierEGA</code> Uses hierarchical exploratory graph analysis • <code>riEGA</code> Uses random-intercept exploratory graph analysis
model	<p>Character. A string indicating the method to use.</p> <p>Current options are:</p> <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
model.args	<p>List. A list of additional arguments for <code>EBICglasso.qgraph</code> or <code>TMFG</code></p>
algorithm	<p>A string indicating the algorithm to use or a function from <code>igraph</code> Defaults to "walktrap". Current options are:</p> <ul style="list-style-type: none"> • <code>walktrap</code> Computes the Walktrap algorithm using <code>cluster_walktrap</code> • <code>leiden</code> Computes the Leiden algorithm using <code>cluster_leiden</code>. Defaults to <code>objective_function = "modularity"</code> • <code>louvain</code> Computes the Louvain algorithm using <code>cluster_louvain</code>
algorithm.args	<p>List. A list of additional arguments for <code>cluster_walktrap</code>, <code>cluster_louvain</code>, or some other community detection algorithm function (see examples)</p>
consensus.method	<p>Character. What consensus clustering method should be used? Defaults to "highest_modularity". Current options are:</p> <ul style="list-style-type: none"> • <code>highest_modularity</code> Uses the community solution that achieves the highest modularity across iterations

	<ul style="list-style-type: none"> • <code>most_common</code> Uses the community solution that is found the most across iterations • <code>iterative</code> Identifies the most common community solutions across iterations and determines how often nodes appear in the same community together. A threshold of 0.30 is used to set low proportions to zero. This process repeats iteratively until all nodes have a proportion of 1 in the community solution. • <code>lowest_tefi</code> Uses the community solution that achieves the lowest <code>tefi</code> across iterations
<code>consensus.iter</code>	Numeric. Number of iterations to perform in consensus clustering for the Louvain algorithm (see Lancichinetti & Fortunato, 2012). Defaults to 100
<code>typicalStructure</code>	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
<code>plot.typicalStructure</code>	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
<code>plot.args</code>	List. A list of additional arguments for the network plot. See <code>ggnet2</code> for full list of arguments: <ul style="list-style-type: none"> • <code>vsize</code> Size of the nodes. Defaults to 6. • <code>label.size</code> Size of the labels. Defaults to 5. • <code>alpha</code> The level of transparency of the nodes, which might be a single value or a vector of values. Defaults to 0.7. • <code>edge.alpha</code> The level of transparency of the edges, which might be a single value or a vector of values. Defaults to 0.4. • <code>legend.names</code> A vector with names for each dimension • <code>color.palette</code> The color palette for the nodes. For custom colors, enter HEX codes for each dimension in a vector. See <code>color_palette_EGA</code> for more details and examples
<code>ncores</code>	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 If you're unsure how many cores your computer has, then use the following code: <code>parallel::detectCores()</code>
<code>progress</code>	Boolean. Should progress be displayed? Defaults to TRUE. For Windows, FALSE is about 2x faster
<code>...</code>	Additional arguments. Used for deprecated arguments from previous versions of EGA

Value

Returns a list containing:

iter	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: <ul style="list-style-type: none"> • 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 Golino <hfg9s at virginia.edu> and Alexander P. Christensen <alexpaulchristensen@gmail.com>

References

Original implementation of bootEGA
Christensen, A. P., & Golino, H. (2021). Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis: A tutorial. *Psych*, 3(3), 479-500.

Structural consistency (see [dimensionStability](#))
Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, 34(6), 1095-1108.

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:
# Standard EGA example
boot.wmt <- bootEGA(
  data = wmt, iter = 500,
  type = "parametric", ncores = 2
)

# Produce Methods section
```

```
methods.section(boot.wmt)

# Louvain example
boot.wmt.louvain <- bootEGA(
  data = wmt, iter = 500,
  algorithm = "louvain",
  type = "parametric", ncores = 2
)

# Spinglass example
boot.wmt.spinglass <- bootEGA(
  data = wmt, iter = 500,
  algorithm = igraph::cluster_spinglass, # use any function from {igraph}
  type = "parametric", ncores = 2
)

# EGA fit example
boot.wmt.fit <- bootEGA(
  data = wmt, iter = 500,
  EGA.type = "EGA.fit",
  type = "parametric", ncores = 2
)

# Hierarchical EGA example
boot.wmt.hier <- bootEGA(
  data = wmt, iter = 500,
  EGA.type = "hierEGA",
  type = "parametric", ncores = 2
)

# Random-intercept EGA example
boot.wmt.ri <- bootEGA(
  data = wmt, iter = 500,
  EGA.type = "riEGA",
  type = "parametric", ncores = 2
)
## End(Not run)
```

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>

References

- Christensen, A. P., Gross, G. M., Golino, H., Silvia, P. J., & Kwapil, T. R. (2019). Exploratory graph analysis of the Multidimensional Schizotypy Scale. *Schizophrenia Research*, 206, 43-51.
- Golino, H., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PLoS ONE*, 12, e0174035.

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,
  plot.EGA = FALSE # No plot for CRAN checks
```

```

)

# Fit CFA model to EGA results
cfa.wmt <- CFA(
  ega.obj = ega.wmt, estimator = "WLSMV",
  plot.CFA = FALSE, # No plot for CRAN checks
  data = wmt
)

# Additional fit measures
lavaan::fitMeasures(cfa.wmt$fit, fit.measures = "all")
## End(Not run)

```

color_palette_EGA [EGA Color Palettes](#)

Description

Color palettes for plotting [ggnet2](#) EGA network plots

Usage

```
color_palette_EGA(name, wc, sorted = FALSE)
```

Arguments

name	<p>Character. Name of color scheme (see RColorBrewer). Defaults to "polychrome".</p> <p>EGA palettes:</p> <ul style="list-style-type: none"> • "polychrome" Default 20 color palette • "grayscale" "grayscale", "greyscale", or "colorblind" will produce plots suitable for publication purposes • "blue.ridge1" Palette inspired by the Blue Ridge Mountains • "blue.ridge2" Second palette inspired by the Blue Ridge Mountains • "rainbow" Rainbow colors. Default for qgraph • "rio" Palette inspired by Rio de Janeiro, Brazil • "itacare" Palette inspired by Itacare, Brazil <p>For custom colors, enter HEX codes for each dimension in a vector</p>
wc	<p>Vector. A vector representing the community (dimension) membership of each node in the network. NA values mean that the node was disconnected from the network</p>
sorted	<p>Boolean. Should colors be sorted by wc? Defaults to TRUE</p>

Value

Vector of colors for community memberships

Author(s)

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

Examples

```
# Default
color_palette_EGA(name = "polychrome", wc = ega.wmt$wc)

# Blue Ridge Mountains 1
color_palette_EGA(name = "blue.ridge1", wc = ega.wmt$wc)

# Custom
color_palette_EGA(name = "#7FD1B9", wc = ega.wmt$wc)
```

compare.EGA.plots *Visually Compares [EGAnet](#) plots*

Description

Organizes EGA plots for comparison. Ensures that nodes are placed in the same layout to maximize comparison. Community memberships are also homogenized across EGA outputs to enhance interpretation

Usage

```
compare.EGA.plots(
  ...,
  input.list = NULL,
  base.plot = 1,
  labels,
  rows,
  columns,
  plot.args = list()
)
```

Arguments

...	EGAnet objects
input.list	List. Bypasses ... argument in favor of using a list as an input
base.plot	Numeric. Plot to be used as the base for the configuration of the networks. Uses the number of the order in which the plots are input. Defaults to 1 or the first plot
labels	Character vector. Labels for each EGAnet object
rows	Numeric. Number of rows to spread plots across
columns	Numeric. Number of columns to spread plots down

`plot.args` List. A list of additional arguments for the network plot. For `plot.type = "qgraph"`:

- `vsize` Size of the nodes. Defaults to 6.

(see [ggnet2](#) for full list of arguments):

- `vsize` Size of the nodes. Defaults to 6.
- `label.size` Size of the labels. Defaults to 5.
- `alpha` The level of transparency of the nodes, which might be a single value or a vector of values. Defaults to 0.7.
- `edge.alpha` The level of transparency of the edges, which might be a single value or a vector of values. Defaults to 0.4.
- `legend.names` A vector with names for each dimension
- `color.palette` The color palette for the nodes. For custom colors, enter HEX codes for each dimension in a vector. See [color_palette_EGA](#) for more details and examples

Value

Visual comparison of [EGAnet](#) objects

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

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

# Draw random samples
sample1 <- items[sample(1:nrow(items), 1000),]
sample2 <- items[sample(1:nrow(items), 1000),]

## Not run:
# Estimate EGAs
ega1 <- EGA(sample1)
ega2 <- EGA(sample2)

# Compare EGAs via plot
compare.EGA.plots(
  ega1, ega2,
  base.plot = 1, # use "ega1" as base for comparison
  labels = c("Sample 1", "Sample 2"),
  rows = 1, columns = 2
)
## End(Not run)
```

convert2igraph	Convert networks to igraph
----------------	--

Description

Converts networks to [igraph](#) format

Usage

```
convert2igraph(A, diagonal = 0)
```

Arguments

A	Matrix or data frame. $N \times N$ matrix where N is the number of nodes
diagonal	Numeric. Value to be placed on the diagonal of A. Defaults to 0

Value

Returns a network in the [igraph](#) format

Author(s)

Hudson Golino <hfg9s at virginia.edu> & Alexander P. Christensen <alexander.christensen at Vanderbilt.Edu>

Examples

```
convert2igraph(ega.wmt$network)
```

depression	<i>Depression Data</i>
------------	------------------------

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")
```

dimensionStability *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

```
dimensionStability(bootega.obj, ...)
```

Arguments

`bootega.obj` A [bootEGA](#) object

`...` Additional arguments. Used for deprecated arguments from previous versions of `dimStability`

Value

Returns a list containing:

`dimension.stability`

A list containing:

- `structural.consistency` The proportion of times that each empirical [EGA](#) dimension *exactly* replicates across the [bootEGA](#) samples
- `average.item.stability` The average item stability in each empirical [EGA](#) dimension

`item.stability` Results from [itemStability](#)

Author(s)

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

References

Christensen, A. P., & Golino, H. (2021). Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis: A tutorial. *Psych*, 3(3), 479-500.

Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, 34(6), 1095-1108.

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 bootstrap EGA
boot.wmt <- bootEGA(
  data = wmt, iter = 500,
  type = "parametric", ncores = 2
)
## End(Not run)

# Estimate stability statistics
res <- dimensionStability(boot.wmt)
res$dimension.stability

## Not run:
# Produce Methods section
methods.section(
  boot.wmt,
  stats = "dimensionStability"
)
## 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 variables less than or equal to factors vs. network model (`hlf_n_weights`), and high correlation with variables greater 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")
```

 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"),
  model.args = list(),
  algorithm = c("walktrap", "leiden", "louvain"),
  algorithm.args = list(),
  corr = c("cor_auto", "pearson", "spearman"),
  uni.method = c("expand", "LE", "louvain"),
  ncores,
  progress = TRUE,
  ...
)
```

Arguments

data	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.)
------	---

n.embed	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.
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".
level	Character. A string indicating the level of analysis. If the interest is in modeling the intraindividual structure only (one dimensionality structure per individual), then level should be set to "individual". If the interest is in the structure of a group of individuals, then level should be set to "group". Finally, if the interest is in the population structure, then level should be set to "population". Current options are: <ul style="list-style-type: none"> • individual 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 ("id") needs to be provided identifying each individual. • group Estimates the dynamic factors for each group. An additional parameter ("group") needs to be provided identifying the group membership. • population Estimates the dynamic factors of the population
id	Numeric. Number of the column identifying each individual.
group	Numeric or character. Number of the column identifying group membership. Must be specified only if level = "group".
use.derivatives	Integer. The order of the derivative to be used in the EGA procedure. Default to 1.
model	Character. A string indicating the method to use. 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. This is the default method • TMFG Estimates a Triangulated Maximally Filtered Graph
model.args	List. A list of additional arguments for EBICglasso.qgraph or TMFG
algorithm	A string indicating the algorithm to use or a function from igraph Defaults to "walktrap". Current options are: <ul style="list-style-type: none"> • walktrap Computes the Walktrap algorithm using cluster_walktrap • leiden Computes the Leiden algorithm using cluster_leiden. Defaults to objective_function = "modularity" • louvain Computes the Louvain algorithm using cluster_louvain
algorithm.args	List. A list of additional arguments for cluster_walktrap , cluster_louvain , or some other community detection algorithm function (see examples)
corr	Type of correlation matrix to compute. The default uses "pearson". 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 <code>qgraph</code>. • <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>uni.method</code>	<p>Character. What unidimensionality method should be used? Defaults to "LE". Current options are:</p> <ul style="list-style-type: none"> • <code>expand</code> Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation. • <code>LE</code> Applies the Leading Eigenvalue algorithm (<code>cluster_leading_eigen</code>) on the empirical correlation matrix. If the number of dimensions is 1, then the Leading Eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation. • <code>louvain</code> Applies the Louvain algorithm (<code>cluster_louvain</code>) on the empirical correlation matrix using a resolution parameter = 0.95. If the number of dimensions is 1, then the Louvain solution is used; otherwise, regular EGA is used. This method was validated in the Christensen (2022) simulation.
<code>ncores</code>	<p>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</p> <p>If you're unsure how many cores your computer has, then use the following code: <code>parallel::detectCores()</code></p>
<code>progress</code>	<p>Boolean. Should progress be displayed? Defaults to TRUE. For Windows, FALSE is about 2x faster</p>
<code>...</code>	<p>Additional arguments. Used for deprecated arguments from previous versions of EGA</p>

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*.
- 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.

Golino, H., Christensen, A. P., Moulder, R. G., Kim, S., & Boker, S. M. (2021). 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. *Psychometrika*.

Savitzky, A., & Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8), 1627-1639.

Examples

```
# Obtain data
sim.dynEGA <- sim.dynEGA # bypasses CRAN checks

## Not run:
# Population structure
dyn.random <- dynEGA(
  data = sim.dynEGA, n.embed = 5, tau = 1,
  delta = 1, id = 25, group = 26, use.derivatives = 1,
  level = "population", ncores = 2, corr = "pearson"
)

# Plot population structure
plot(dyn.random)

# Group structure
dyn.group <- dynEGA(
  data = sim.dynEGA, n.embed = 5, tau = 1,
  delta = 1, id = 25, group = 26, use.derivatives = 1,
  level = "group", ncores = 2, corr = "pearson"
)

# Plot group structure
plot(dyn.group, ncol = 2, nrow = 1)

# Intraindividual structure
dyn.individual <- dynEGA(
  data = sim.dynEGA, n.embed = 5, tau = 1,
  delta = 1, id = 25, group = 26, use.derivatives = 1,
  level = "individual", ncores = 2, corr = "pearson"
)

# Plot individual structure (participant 1)
plot(dyn.individual, id = 1)
## End(Not run)
```

Description

DynEGA 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) and 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.ind.pop(
  data,
  n.embed,
  tau = 1,
  delta = 1,
  id = NULL,
  use.derivatives = 1,
  model = c("glasso", "TMFG"),
  model.args = list(),
  algorithm = c("walktrap", "leiden", "louvain"),
  algorithm.args = list(),
  corr = c("cor_auto", "pearson", "spearman"),
  uni.method = c("expand", "LE", "louvain"),
  ncores,
  progress = TRUE,
  ...
)
```

Arguments

<code>data</code>	A data frame with the variables to be used in the analysis. The data frame 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 <code>n.embed = 5</code> 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 <code>tau</code> of one uses adjacent observations. Default is <code>tau = 1</code> .
<code>delta</code>	Integer. The time between successive observations in the time series. Default is <code>delta = 1</code> .
<code>id</code>	Numeric. Number of the column identifying each individual.
<code>use.derivatives</code>	Integer. The order of the derivative to be used in the EGA procedure. Default to 1.

model	<p>Character. A string indicating the method to use. Defaults to <code>glasso</code>. Current options are:</p> <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
model.args	List. A list of additional arguments for <code>EBICglasso.qgraph</code> or <code>TMFG</code>
algorithm	<p>A string indicating the algorithm to use or a function from <code>igraph</code> Defaults to <code>"walktrap"</code>. Current options are:</p> <ul style="list-style-type: none"> • <code>walktrap</code> Computes the Walktrap algorithm using <code>cluster_walktrap</code> • <code>leiden</code> Computes the Leiden algorithm using <code>cluster_leiden</code>. Defaults to <code>objective_function = "modularity"</code> • <code>louvain</code> Computes the Louvain algorithm using <code>cluster_louvain</code>
algorithm.args	List. A list of additional arguments for <code>cluster_walktrap</code> , <code>cluster_louvain</code> , or some other community detection algorithm function (see examples)
corr	<p>Type of correlation matrix to compute. The default uses <code>cor_auto</code>. Current options are:</p> <ul style="list-style-type: none"> • <code>cor_auto</code> Computes the correlation matrix using the <code>cor_auto</code> function from <code>qgraph</code>. • <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.
uni.method	<p>Character. What unidimensionality method should be used? Defaults to <code>"LE"</code>. Current options are:</p> <ul style="list-style-type: none"> • <code>expand</code> Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation. • <code>LE</code> Applies the Leading Eigenvalue algorithm (<code>cluster_leading_eigen</code>) on the empirical correlation matrix. If the number of dimensions is 1, then the Leading Eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation. • <code>louvain</code> Applies the Louvain algorithm (<code>cluster_louvain</code>) on the empirical correlation matrix using a resolution parameter = 0.95. If the number of dimensions is 1, then the Louvain solution is used; otherwise, regular EGA is used. This method was validated in the Christensen (2022) simulation.
ncores	<p>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</p> <p>If you're unsure how many cores your computer has, then use the following code: <code>parallel::detectCores()</code></p>

progress Boolean. Should progress be displayed? Defaults to TRUE. For Windows, FALSE is about 2x faster

... Additional arguments. Used for deprecated arguments from previous versions of [EGA](#)

Author(s)

Hudson Golino <hfg9s at virginia.edu>

Examples

```
# Obtain data
sim.dynEGA <- sim.dynEGA # bypasses CRAN checks

## Not run:
# Dynamic EGA individual and population structure
dyn.ega1 <- dynEGA.ind.pop(
  data = sim.dynEGA, n.embed = 5, tau = 1,
  delta = 1, id = 25, use.derivatives = 1,
  ncores = 2, corr = "pearson"
)
## 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 = 0.1,
  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. Defaults to 0.1. NOTE <code>qgraph</code> sets the default to 0.01
<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 <code>qgraph < 1.3</code> behavior (not recommended!).
<code>refit</code>	Logical, should the optimal graph be refitted without LASSO regularization? Defaults to FALSE.
<code>...</code>	Arguments sent to <code>glasso</code>

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

- ```
Instantiation of GLASSO
Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9, 432-441.
Tutorial on EBICglasso Epskamp, S., & Fried, E. I. (2018). A tutorial on regularized partial correlation networks. Psychological Methods, 23(4), 617-634.
glasso package
Friedman, J., Hastie, T., & Tibshirani, R. (2011). glasso: Graphical lasso-estimation of Gaussian graphical models. R package version 1.7.
```

```
glasso + EBIC
Foygel, R., & Drton, M. (2010). Extended Bayesian information criteria for Gaussian graphical models. In Advances in neural information processing systems (pp. 604-612).
```

### Examples

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

Not run:
Compute graph with tuning = 0 (BIC)
BICgraph <- EBICglasso.qgraph(
 data = wmt, gamma = 0
)

Compute graph with tuning = 0.5 (EBIC)
EBICgraph <- EBICglasso.qgraph(
 data = wmt, gamma = 0.5
)
End(Not run)
```

---

 EGA

*Applies the Exploratory Graph Analysis technique*


---

### Description

Estimates the number of dimensions of a given dataset or correlation matrix using the graphical lasso ([EBICglasso.qgraph](#)) or the Triangulated Maximally Filtered Graph ([TMFG](#)) network estimation methods.

### Usage

```
EGA(
 data,
 n = NULL,
 corr = c("cor_auto", "pearson", "spearman"),
 uni.method = c("expand", "LE", "louvain"),
 model = c("glasso", "TMFG"),
 model.args = list(),
 algorithm = c("walktrap", "leiden", "louvain"),
 algorithm.args = list(),
 consensus.method = c("highest_modularity", "most_common", "iterative", "lowest_tefi"),
 consensus.iter = 100,
 plot.EGA = TRUE,
 plot.args = list(),
 ...
)
```

**Arguments**

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>data</code>           | Matrix or data frame. Variables (down columns) or correlation matrix. If the input is a correlation matrix, then argument <code>n</code> (number of cases) is <b>required</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>n</code>              | Integer. Sample size if <code>data</code> provided is a correlation matrix                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>corr</code>           | Type of correlation matrix to compute. The default uses <code>cor_auto</code> . Current options are: <ul style="list-style-type: none"> <li>• <code>cor_auto</code> Computes the correlation matrix using the <code>cor_auto</code> function from <code>qgraph</code>.</li> <li>• <code>pearson</code> Computes Pearson's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> <li>• <code>spearman</code> Computes Spearman's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>uni.method</code>     | Character. What unidimensionality method should be used? Defaults to "louvain". Current options are: <ul style="list-style-type: none"> <li>• <code>expand</code> Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation.</li> <li>• <code>LE</code> Applies the Leading Eigenvalue algorithm (<code>cluster_leading_eigen</code>) on the empirical correlation matrix. If the number of dimensions is 1, then the Leading Eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation.</li> <li>• <code>louvain</code> Applies the Louvain algorithm (<code>cluster_louvain</code>) on the empirical correlation matrix using a resolution parameter = 0.95. If the number of dimensions is 1, then the Louvain solution is used; otherwise, regular EGA is used. This method was validated in the Christensen (2022) simulation.</li> </ul> |
| <code>model</code>          | Character. A string indicating the method to use. Defaults to "glasso". Current options are: <ul style="list-style-type: none"> <li>• <code>glasso</code> Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter</li> <li>• <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>model.args</code>     | List. A list of additional arguments for <code>EBICglasso.qgraph</code> or <code>TMFG</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>algorithm</code>      | A string indicating the algorithm to use or a function from <code>igraph</code> Defaults to "walktrap". Current options are: <ul style="list-style-type: none"> <li>• <code>walktrap</code> Computes the Walktrap algorithm using <code>cluster_walktrap</code></li> <li>• <code>leiden</code> Computes the Leiden algorithm using <code>cluster_leiden</code>. Defaults to <code>objective_function = "modularity"</code></li> <li>• <code>louvain</code> Computes the Louvain algorithm using <code>cluster_louvain</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>algorithm.args</code> | List. A list of additional arguments for <code>cluster_walktrap</code> , <code>cluster_louvain</code> , or some other community detection algorithm function (see examples)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>consensus.method</code> | <p>Character. What consensus clustering method should be used? Defaults to "highest_modularity". Current options are:</p> <ul style="list-style-type: none"> <li>• <code>highest_modularity</code> Uses the community solution that achieves the highest modularity across iterations</li> <li>• <code>most_common</code> Uses the community solution that is found the most across iterations</li> <li>• <code>iterative</code> Identifies the most common community solutions across iterations and determines how often nodes appear in the same community together. A threshold of 0.30 is used to set low proportions to zero. This process repeats iteratively until all nodes have a proportion of 1 in the community solution.</li> <li>• <code>lowest_tefi</code> Uses the community solution that achieves the lowest <code>tefi</code> across iterations</li> </ul>                                                                                                                                                                                                          |
| <code>consensus.iter</code>   | Numeric. Number of iterations to perform in consensus clustering for the Louvain algorithm (see Lancichinetti & Fortunato, 2012). Defaults to 100                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>plot.EGA</code>         | Boolean. If TRUE, returns a plot of the network and its estimated dimensions. Defaults to TRUE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>plot.args</code>        | <p>List. A list of additional arguments for the network plot. For <code>plot.type = "qgraph"</code>:</p> <ul style="list-style-type: none"> <li>• <code>vsize</code> Size of the nodes. Defaults to 6.</li> </ul> <p>For <code>plot.type = "GGally"</code> (see <code>ggnet2</code> for full list of arguments):</p> <ul style="list-style-type: none"> <li>• <code>vsize</code> Size of the nodes. Defaults to 6.</li> <li>• <code>label.size</code> Size of the labels. Defaults to 5.</li> <li>• <code>alpha</code> The level of transparency of the nodes, which might be a single value or a vector of values. Defaults to 0.7.</li> <li>• <code>edge.alpha</code> The level of transparency of the edges, which might be a single value or a vector of values. Defaults to 0.4.</li> <li>• <code>legend.names</code> A vector with names for each dimension</li> <li>• <code>color.palette</code> The color palette for the nodes. For custom colors, enter HEX codes for each dimension in a vector. See <code>color_palette_EGA</code> for more details and examples</li> </ul> |
| <code>...</code>              | Additional arguments. Used for deprecated arguments from previous versions of <code>EGA</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

## Details

Two community detection algorithms, Walktrap (Pons & Latapy, 2006) and Louvain (Blondel et al., 2008), are pre-programmed because of their superior performance in simulation studies on psychological data generated from factor models (Christensen & Golino; 2020; Golino et al., 2020). Notably, any community detection algorithm from the `igraph` can be used to estimate the number of communities (see examples).

**Value**

Returns a list containing:

|          |                                                                                                                                                        |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| network  | A symmetric network estimated using either the <a href="#">EBICglasso.qgraph</a> or <a href="#">TMFG</a>                                               |
| 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)**

Hudson 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**

- # Louvain algorithm  
Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008, P10008.
- # Comprehensive unidimensionality simulation  
Christensen, A. P. (2022). Unidimensional community detection: A Monte Carlo simulation, grid search, and comparison. *PsyArXiv*.
- # Compared all *igraph* community detections algorithms, introduced Louvain algorithm, simulation with continuous and polytomous data  
# Also implements the Leading Eigenvalue unidimensional method  
Christensen, A. P., Garrido, L. E., & Golino, H. (2021). Comparing community detection algorithms in psychological data: A Monte Carlo simulation. *PsyArXiv*.
- # Original simulation and implementation of EGA  
Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PLoS ONE*, 12, e0174035.
- Golino, H. F., & Demetriou, A. (2017). Estimating the dimensionality of intelligence like data using Exploratory Graph Analysis. *Intelligence*, 62, 54-70.
- # Current implementation of EGA, introduced unidimensional checks, continuous and dichotomous data  
Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., & Thiyagarajan, J. A. (2020). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, 25, 292-320.
- # Walktrap algorithm  
Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, 10, 191-218.

**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

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

Not run:
Estimate EGA
ega.wmt <- EGA(
 data = wmt,
 plot.EGA = FALSE # No plot for CRAN checks
)

Summary statistics
summary(ega.wmt)

Produce Methods section
methods.section(ega.wmt)

Estimate EGAtmfg
ega.wmt.tmfg <- EGA(
 data = wmt, model = "TMFG"
)

Estimate EGA with Louvain algorithm
ega.wmt.louvain <- EGA(
 data = wmt, algorithm = "louvain"
)

Estimate EGA with Leiden algorithm
ega.wmt.leiden <- EGA(
 data = wmt, algorithm = "leiden"
)

Estimate EGA with Spinglass algorithm
ega.wmt.spinglass <- EGA(
 data = wmt,
 algorithm = igraph::cluster_spinglass
)
End(Not run)
```

---

EGA.estimate

*A Sub-routine Function for EGA*

---

## Description

Estimates the number of dimensions of a given dataset or correlation matrix using the graphical lasso ([EBICglasso.qgraph](#)) or the Triangulated Maximally Filtered Graph ([TMFG](#)) network estimation methods.



**Usage**

```
EGA.estimate(
 data,
 n = NULL,
 corr = c("cor_auto", "pearson", "spearman"),
 model = c("glasso", "TMFG"),
 model.args = list(),
 algorithm = c("walktrap", "leiden", "louvain"),
 algorithm.args = list(),
 consensus.method = c("highest_modularity", "most_common", "iterative", "lowest_tefi"),
 consensus.iter = 100,
 ...
)
```

**Arguments**

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data             | Matrix or data frame. Variables (down columns) or correlation matrix. If the input is a correlation matrix, then argument n (number of cases) is <b>required</b>                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| n                | Integer. Sample size if data provided is a correlation matrix                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| corr             | Type of correlation matrix to compute. The default uses <code>cor_auto</code> . Current options are: <ul style="list-style-type: none"> <li>• <code>cor_auto</code> Computes the correlation matrix using the <code>cor_auto</code> function from <code>qgraph</code>.</li> <li>• <code>pearson</code> Computes Pearson's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> <li>• <code>spearman</code> Computes Spearman's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> </ul> |
| model            | Character. A string indicating the method to use. Current options are: <ul style="list-style-type: none"> <li>• <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</li> <li>• <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph</li> </ul>                                                                                                                                                                                    |
| model.args       | List. A list of additional arguments for <code>EBICglasso.qgraph</code> or <code>TMFG</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| algorithm        | A string indicating the algorithm to use or a function from <code>igraph</code> Current options are: <ul style="list-style-type: none"> <li>• <code>walktrap</code> Computes the Walktrap algorithm using <code>cluster_walktrap</code></li> <li>• <code>leiden</code> Computes the Leiden algorithm using <code>cluster_leiden</code></li> <li>• <code>louvain</code> Computes the Louvain algorithm using <code>cluster_louvain</code></li> </ul>                                                                                                                                               |
| algorithm.args   | List. A list of additional arguments for <code>cluster_walktrap</code> , <code>cluster_louvain</code> , or some other community detection algorithm function (see examples)                                                                                                                                                                                                                                                                                                                                                                                                                       |
| consensus.method | Character. What consensus clustering method should be used? Defaults to <code>"highest_modularity"</code> . Current options are:                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

- `highest_modularity` Uses the community solution that achieves the highest modularity across iterations
  - `most_common` Uses the community solution that is found the most across iterations
  - `iterative` Identifies the most common community solutions across iterations and determines how often nodes appear in the same community together. A threshold of 0.30 is used to set low proportions to zero. This process repeats iteratively until all nodes have a proportion of 1 in the community solution.
  - `lowest_tefi` Uses the community solution that achieves the lowest `tefi` across iterations
- `consensus.iter` Numeric. Number of iterations to perform in consensus clustering for the Louvain algorithm (see Lancichinetti & Fortunato, 2012). Defaults to 100
- ... Additional arguments. Used for deprecated arguments from previous versions of EGA

### Details

Two community detection algorithms, Walktrap (Pons & Latapy, 2006) and Louvain (Blondel et al., 2008), are pre-programmed because of their superior performance in simulation studies on psychological data generated from factor models (Christensen & Golino; 2020; Golino et al., 2020). Notably, any community detection algorithm from the `igraph` can be used to estimate the number of communities (see examples).

### 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 at gmail.com> and Hudson Golino <hfg9s at virginia.edu>

### References

- # Louvain algorithm  
Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008, P10008.
- # Compared all `igraph` community detections algorithms, introduced Louvain algorithm, simulation with continuous and polytomous data

Christensen, A. P., & Golino, H. (under review). Estimating factors with psychometric networks: A Monte Carlo simulation comparing community detection algorithms. *PsyArXiv*.

# Original simulation and implementation of EGA

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PLoS ONE*, *12*, e0174035.

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

# Current implementation of EGA, introduced unidimensional checks, continuous and dichotomous data

Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., & Thiagarajan, J. A. (2020). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, *25*, 292-320.

# Walktrap algorithm

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

## 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

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

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

Estimate EGAtmfg
ega.wmt.tmfg <- EGA.estimate(data = wmt, model = "TMFG")

Estimate EGA with Louvain algorithm
ega.wmt.louvain <- EGA.estimate(data = wmt, algorithm = "louvain")

Estimate EGA with Spinglass algorithm
ega.wmt.spinglass <- EGA.estimate(
 data = wmt,
 algorithm = igraph::cluster_spinglass # any {igraph} algorithm
)
End(Not run)
```

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](#).

## Usage

```
EGA.fit(
 data,
 n = NULL,
 uni.method = c("expand", "LE"),
 corr = c("cor_auto", "pearson", "spearman"),
 model = c("glasso", "TMFG"),
 algorithm = c("leiden", "walktrap"),
 algorithm.args = list(steps = c(3:8), resolution_parameter = seq(0, 2, 0.001))
)
```

## Arguments

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data       | Matrix or data frame. Dataset or correlation matrix                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| n          | Integer. Sample size (if the data provided is a correlation matrix)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| uni.method | Character. What unidimensionality method should be used? Defaults to "LE". Current options are: <ul style="list-style-type: none"> <li>• expand Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation.</li> <li>• LE Applies the leading eigenvalue algorithm (<a href="#">cluster_leading_eigen</a>) on the empirical correlation matrix. If the number of dimensions is 1, then the leading eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation.</li> </ul> |
| corr       | Type of correlation matrix to compute. The default uses <a href="#">cor_auto</a> . Current options are: <ul style="list-style-type: none"> <li>• cor_auto Computes the correlation matrix using the <a href="#">cor_auto</a> function from <a href="#">qgraph</a>.</li> <li>• pearson Computes Pearson's correlation coefficient using the pairwise complete observations via the <a href="#">cor</a> function.</li> <li>• spearman Computes Spearman's correlation coefficient using the pairwise complete observations via the <a href="#">cor</a> function.</li> </ul>                                                                                                                                                                                                                                             |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| model          | Character. A string indicating the method to use. Defaults to "glasso"<br>Current options are: <ul style="list-style-type: none"> <li>• "glasso" Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter. See <a href="#">EBICglasso.qgraph</a></li> <li>• "TMFG" Estimates a Triangulated Maximally Filtered Graph. See <a href="#">TMFG</a></li> </ul> |
| algorithm      | A string indicating the algorithm to use or a function from <a href="#">igraph</a> Defaults to "walktrap". Current options are: <ul style="list-style-type: none"> <li>• walktrap Computes the Walktrap algorithm using <a href="#">cluster_walktrap</a></li> <li>• leiden Computes the Leiden algorithm using <a href="#">cluster_louvain</a></li> </ul>                                                                                             |
| algorithm.args | List. A list of additional arguments for <a href="#">cluster_walktrap</a> or <a href="#">cluster_leiden</a> . Options are: <ul style="list-style-type: none"> <li>• steps Number of steps used in the Walktrap algorithm. Defaults to c(3:8)</li> <li>• leiden Resolution parameter used in the Leiden algorithm. Defaults to seq(0, 2, .001). Higher values lead to smaller communities, lower values lead to larger communities</li> </ul>          |

### Value

Returns a list containing:

|                      |                                                                                                                               |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------|
| EGA                  | The <a href="#">EGA</a> output for the best fitting model                                                                     |
| steps                | The number of steps used in the best fitting model from the <a href="#">cluster_walktrap</a> algorithm                        |
| resolution_parameter | The resolution parameter used in the best fitting model from the <a href="#">cluster_leiden</a> algorithm                     |
| EntropyFit           | The <a href="#">tefi</a> Index for the unique solutions given the range of steps (vector names represent the number of steps) |
| Lowest.EntropyFit    | The lowest value for the <a href="#">tefi</a> Index                                                                           |

### Author(s)

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

### References

# Entropy fit measures

Golino, H., Moulder, R. G., Shi, D., Christensen, A. P., Garrido, L. E., Neito, M. D., Nesselroade, J., Sadana, R., Thiagarajan, J. A., & Boker, S. M. (in press). Entropy fit indices: New fit measures for assessing the structure and dimensionality of multiple latent variables. *Multivariate Behavioral Research*.

# Simulation for EGA.fit

Jamison, L., Christensen, A. P., & Golino, H. (under review). Optimizing Walktrap's community detection in networks using the Total Entropy Fit Index. *PsyArXiv*.

```
Leiden algorithm
Traag, V. A., Waltman, L., & Van Eck, N. J. (2019). From Louvain to Leiden: guaranteeing well-
connected communities. Scientific Reports, 9(1), 1-12.

Walktrap algorithm
Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks.
Journal of Graph Algorithms and Applications, 10, 191-218.
```

### 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 EGA
ega.wmt <- EGA(
 data = wmt,
 plot.EGA = FALSE # No plot for CRAN checks
)

Estimate optimal EGA
fit.wmt <- EGA.fit(data = wmt)

Plot optimal fit
plot(fit.wmt$EGA)

Estimate CFAs
cfa.ega <- CFA(ega.wmt, estimator = "WLSMV", data = wmt)
cfa.fit <- CFA(fit.wmt$EGA, estimator = "WLSMV", data = wmt)

Compare CFAs
lavaan::lavTestLRT(
 cfa.ega$fit, cfa.fit$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")
```

---

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*, 367-386.

## 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 <a href="#">EGA</a> |

## 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., Moulder, R. G., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Nesselroade, J., Sadana, R., Thiyagarajan, J. A., & Boker, S. M. (2020). Entropy fit indices: New fit measures for assessing the structure and dimensionality of multiple latent variables. *Multivariate Behavioral Research*.

**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)
End(Not run)

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

---

ergoInfo

*Ergodicity Information Index*

---

**Description**

Computes the Ergodicity Information Index

**Usage**

```
ergoInfo(dynEGA.object, use = c("edge.list", "unweighted", "weighted"))
```

**Arguments**

`dynEGA.object` A [dynEGA.ind.pop](#) object

`use` Character. A string indicating what network element will be used to compute the algorithm complexity, the list of edges or the weights of the network. Defaults to `use = "edge.list"`. Current options are:

- "edge.list" Calculates the algorithm complexity using the list of edges.
- "unweighted" Calculates the algorithm complexity using the binary weights of the network. 0 = edge absent and 1 = edge present
- "weighted" Calculates the algorithm complexity using the weights of the network.

### Value

Returns a list containing:

|                 |                                                                           |
|-----------------|---------------------------------------------------------------------------|
| PrimeWeight     | The prime-weight encoding of the individual networks                      |
| PrimeWeight.pop | The prime-weight encoding of the population network                       |
| Kcomp           | The Kolmogorov complexity of the prime-weight encoded individual networks |
| Kcomp.pop       | The Kolmogorov complexity of the prime-weight encoded population network  |
| complexity      | The complexity metric proposed by Santora and Nicosia (2020)              |
| EII             | The Ergodicity Information Index                                          |

### Author(s)

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

### Examples

```
Obtain data
sim.dynEGA <- sim.dynEGA # bypasses CRAN checks

Not run:
Dynamic EGA individual and population structure
dyn.ega1 <- dynEGA.ind.pop(
 data = sim.dynEGA[,-26], n.embed = 5, tau = 1,
 delta = 1, id = 25, use.derivatives = 1,
 ncores = 2, corr = "pearson"
)

Compute empirical ergodicity information index
eii <- ergoInfo(
 dynEGA.object = dyn.ega1,
 use = "weighted"
)
End(Not run)
```

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 <a href="#">Embed</a> function).                                                                                                                                                                                                                             |
| tau     | Integer. Number of observations to offset successive embeddings in the <a href="#">Embed</a> 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 <a href="#">Embed</a> 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*.

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.

Savitzky, A., & Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8), 1627-1639.

## Examples

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

---

hierEGA

*Hierarchical* [EGA](#)

---

## Description

Estimates EGA using the lower-order solution of [cluster\\_louvain](#) to identify the lower-order dimensions and then uses factor or network loadings to estimate factor or network scores, which are used to estimate the higher-order dimensions

## Usage

```
hierEGA(
 data,
 scores = c("factor", "network"),
 consensus.iter = 1000,
 consensus.method = c("highest_modularity", "most_common", "iterative", "lowest_tefi"),
 uni.method = c("expand", "LE", "louvain"),
 corr = c("cor_auto", "pearson", "spearman"),
 model = c("glasso", "TMFG"),
 model.args = list(),
 algorithm = c("walktrap", "leiden", "louvain"),
 algorithm.args = list(),
 plot.EGA = TRUE,
 plot.args = list()
)
```

## Arguments

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data   | Matrix or data frame. Variables (down columns) only. Does not accept correlation matrices                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| scores | Character. How should scores for the higher-order structure be estimated? Defaults to "network" for network scores computed using the <a href="#">net.scores</a> function. Set to "factor" for factor scores computed using <a href="#">fa</a> . Factors are assumed to be correlated using the "oblimin" rotation. <i>NOTE</i> : Factor scores use the number of communities from <a href="#">EGA</a> . Estimated factor may not align with these communities. The plots using factor scores with have higher order factors that may not completely map onto the lower order communities. Look at |

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                               | <p>the <code>\$hierarchical\$higher_order\$lower_loadings</code> to determine the composition of the lower order factors.</p> <p>By default, both factor and network scores are computed and stored in the output. The selected option only appears in the main output (<code>\$hierarchical</code>)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>consensus.iter</code>   | <p>Numeric. Number of iterations to perform in consensus clustering (see Lancichinetti &amp; Fortunato, 2012). Defaults to 1000</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>consensus.method</code> | <p>Character. What consensus clustering method should be used? Defaults to "highest_modularity". Current options are:</p> <ul style="list-style-type: none"> <li>• <code>highest_modularity</code> Uses the community solution that achieves the highest modularity across iterations</li> <li>• <code>most_common</code> Uses the community solution that is found the most across iterations</li> <li>• <code>iterative</code> Identifies the most common community solutions across iterations and determines how often nodes appear in the same community together. A threshold of 0.30 is used to set low proportions to zero. This process repeats iteratively until all nodes have a proportion of 1 in the community solution.</li> <li>• <code>lowest_tefi</code> Uses the community solution that achieves the lowest <code>tefi</code> across iterations</li> </ul> <p>By default, all <code>consensus.method</code> options are computed and stored in the output. The selected method will be used to plot and appear in the main output (<code>\$hierarchical</code>)</p>                                                                                                        |
| <code>uni.method</code>       | <p>Character. What unidimensionality method should be used? Defaults to "LE". Current options are:</p> <ul style="list-style-type: none"> <li>• <code>expand</code> Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation.</li> <li>• <code>LE</code> Applies the Leading Eigenvalue algorithm (<code>cluster_leading_eigen</code>) on the empirical correlation matrix. If the number of dimensions is 1, then the Leading Eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation.</li> <li>• <code>louvain</code> Applies the Louvain algorithm (<code>cluster_louvain</code>) on the empirical correlation matrix using a resolution parameter = 0.95. If the number of dimensions is 1, then the Louvain solution is used; otherwise, regular EGA is used. This method was validated in the Christensen (2022) simulation.</li> </ul> |
| <code>corr</code>             | <p>Type of correlation matrix to compute. The default uses <code>cor_auto</code>. Current options are:</p> <ul style="list-style-type: none"> <li>• <code>cor_auto</code> Computes the correlation matrix using the <code>cor_auto</code> function from <code>qgraph</code>.</li> <li>• <code>pearson</code> Computes Pearson's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | <ul style="list-style-type: none"> <li>• <code>spearman</code> Computes Spearman's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>model</code>          | <p>Character. A string indicating the method to use. Defaults to "glasso". Current options are:</p> <ul style="list-style-type: none"> <li>• <code>glasso</code> Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter</li> <li>• <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>model.args</code>     | List. A list of additional arguments for <code>EBICglasso.qgraph</code> or <code>TMFG</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>algorithm</code>      | <p>A string indicating the algorithm to use or a function from <code>igraph</code> Defaults to "walktrap". Current options are:</p> <ul style="list-style-type: none"> <li>• <code>walktrap</code> Computes the Walktrap algorithm using <code>cluster_walktrap</code></li> <li>• <code>leiden</code> Computes the Leiden algorithm using <code>cluster_leiden</code>. Defaults to <code>objective_function = "modularity"</code></li> <li>• <code>louvain</code> Computes the Louvain algorithm using <code>cluster_louvain</code></li> </ul>                                                                                                                                                                                                                                                                                                                                               |
| <code>algorithm.args</code> | List. A list of additional arguments for <code>cluster_walktrap</code> , <code>cluster_louvain</code> , or some other community detection algorithm function (see examples)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>plot.EGA</code>       | Boolean. If TRUE, returns a plot of the network and its estimated dimensions. Defaults to TRUE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>plot.args</code>      | <p>List. A list of additional arguments for the network plot. See <code>ggnetwork2</code> for full list of arguments:</p> <ul style="list-style-type: none"> <li>• <code>vsize</code> Size of the nodes. Defaults to 6.</li> <li>• <code>label.size</code> Size of the labels. Defaults to 5.</li> <li>• <code>alpha</code> The level of transparency of the nodes, which might be a single value or a vector of values. Defaults to 0.7.</li> <li>• <code>edge.alpha</code> The level of transparency of the edges, which might be a single value or a vector of values. Defaults to 0.4.</li> <li>• <code>legend.names</code> A vector with names for each dimension</li> <li>• <code>color.palette</code> The color palette for the nodes. For custom colors, enter HEX codes for each dimension in a vector. See <code>color_palette_EGA</code> for more details and examples</li> </ul> |

## Value

Returns a list of lists containing:

### Main Results

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>hierarhical</code> | <p>The main results list containing:</p> <ul style="list-style-type: none"> <li>• <code>lower_order</code> Lower order <code>EGA</code> results for the selected methods</li> <li>• <code>higher_order</code> Higher order <code>EGA</code> results for the selected methods</li> </ul> <p>If <code>plot.EGA = TRUE</code>, then:</p> <ul style="list-style-type: none"> <li>• <code>lower_plot</code> Plot of the lower order results</li> </ul> |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- `higher_plot` Plot of the higher order results
- `hier_plot` Plot of the lower and higher order results together, side-by-side

## Secondary Results

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>lower_ega</code>       | A list containing the lower order <a href="#">EGA</a> results. The <code>\$wc</code> does not contain valid results. Do not use its output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>lower_wc</code>        | A list containing consensus clustering results: <ul style="list-style-type: none"> <li>• <code>highest_modularity</code> Community memberships based on the highest modularity across the <a href="#">cluster_louvain</a> applications</li> <li>• <code>most_common</code> Community memberships based on the most commonly found memberships across the <a href="#">cluster_louvain</a> applications</li> <li>• <code>iterative</code> Community memberships based on consensus clustering described by Lancichinetti &amp; Fortunato (2012)</li> <li>• <code>lowest_tefi</code> Community memberships based on the lowest <a href="#">tefi</a> across the <a href="#">cluster_louvain</a> applications</li> <li>• <code>summary_table</code> A data frame summarizing the unique community solutions across the iterations. Down the columns indicate: number of dimensions (<code>N_Dimensions</code>), proportion of times each community solution was identified (<code>Proportion</code>), modularity of each community solution (<code>Modularity</code>), total entropy fit index of each community solution (<code>tefi</code>), and the memberships for each item. Across the rows indicate each unique community solution</li> </ul> |
| <code>factor_results</code>  | A list containing higher order results based on factor scores. A list for each <code>consensus.method</code> is provided with their <a href="#">EGA</a> results                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>network_results</code> | A list containing higher order results based on network scores. A list for each <code>consensus.method</code> is provided with their <a href="#">EGA</a> results                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## Author(s)

Marcos Jimenez <marcosjnezhquez@gmail.com>, Francisco J. Abad <fjose.abad@uam.es>, Eduardo Garcia-Garzon <egarcia@ucjc.edu>, Hudson Golino <hfg9s@virginia.edu>, Alexander P. Christensen <alexpaulchristensen@gmail.com>, and Luis Eduardo Garrido <luisgarrido@pucmm.edu.do>

## References

Lancichinetti, A., & Fortunato, S. (2012). Consensus clustering in complex networks. *Scientific Reports*, 2(1), 1-7.

## Examples

```
Obtain example data
data <- optimism

Not run:
hierEGA example
```

```

opt.hier<- hierEGA(
 data = optimism,
 algorithm = "louvain"
)
End(Not run)

```

---

infoCluster

*Information Theoretic Mixture Clustering for dynEGA*


---

### Description

Performs hierarchical clustering using Jensen-Shannon distance followed by the Louvain algorithm with consensus clustering. The method iteratively identifies smaller and smaller clusters until there is no change in the clusters identified

### Usage

```
infoCluster(dynEGA.object, plot.cluster = TRUE)
```

### Arguments

|                            |                                                                                                                            |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <code>dynEGA.object</code> | A <a href="#">dynEGA</a> or a <a href="#">dynEGA.ind.pop</a> object that is used to match the arguments of the EII object. |
| <code>plot.cluster</code>  | Boolean. Should plot of optimal and hierarchical clusters be output? Defaults to TRUE. Set to FALSE to not plot            |

### Value

Returns a list containing:

|                          |                                                                      |
|--------------------------|----------------------------------------------------------------------|
| <code>clusters</code>    | A vector corresponding to cluster each participant belongs to        |
| <code>clusterTree</code> | The dendogram from <a href="#">hclust</a> the hierarhical clustering |
| <code>clusterPlot</code> | Plot output from results                                             |
| <code>JSD</code>         | Jensen-Shannon Distance                                              |

### Author(s)

Hudson Golino <hfg9s at virginia.edu> & Alexander P. Christensen <alexander.christensen at Vanderbilt.Edu>



**Examples**

```
Obtain data
sim.dynEGA <- sim.dynEGA # bypasses CRAN checks

Not run:
Dynamic EGA individual and population structure
dyn.ega1 <- dynEGA.ind.pop(
 data = sim.dynEGA, n.embed = 5, tau = 1,
 delta = 1, id = 25, use.derivatives = 1,
 ncores = 2, corr = "pearson"
)

Perform information-theoretic clustering
clust1 <- infoCluster(dynEGA.object = dyn.ega1)
End(Not run)
```

---

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")
```

---

 invariance
 

---



---

*Measurement Invariance of [EGA](#) Structure*


---

## Description

Estimates metric invariance of [EGA](#) or specified structure

## Usage

```
invariance(
 data,
 groups,
 iter = 500,
 memberships = NULL,
 type = c("loadings"),
 corr = c("cor_auto", "pearson", "spearman"),
 uni.method = c("expand", "LE", "louvain"),
 model = c("glasso", "TMFG"),
 model.args = list(gamma = 0),
 algorithm = c("walktrap", "leiden", "louvain"),
 algorithm.args = list(),
 consensus.method = c("highest_modularity", "most_common", "iterative", "lowest_tefi"),
 consensus.iter = 100,
 ncores,
 progress = TRUE
)
```

## Arguments

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data        | Matrix or data frame. Variables to be used in the analysis                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| groups      | Vector. Group membership corresponding to each case in data                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| iter        | Numeric. Number of iterations to perform for the permutation. Defaults to 500                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| memberships | Vector. Node membership for each community or factor. Defaults to NULL. When NULL, <a href="#">EGA</a> is used to compute node memberships                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| type        | Character. Type of measurement invariance to estimate. Only includes "loadings" at the moment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| corr        | Type of correlation matrix to compute. The default uses <a href="#">cor_auto</a> . Current options are: <ul style="list-style-type: none"> <li>• <a href="#">cor_auto</a> Computes the correlation matrix using the <a href="#">cor_auto</a> function from <a href="#">qgraph</a>.</li> <li>• <a href="#">pearson</a> Computes Pearson's correlation coefficient using the pairwise complete observations via the <a href="#">cor</a> function.</li> <li>• <a href="#">spearman</a> Computes Spearman's correlation coefficient using the pairwise complete observations via the <a href="#">cor</a> function.</li> </ul> |

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>uni.method</code>       | <p>Character. What unidimensionality method should be used? Defaults to "louvain". Current options are:</p> <ul style="list-style-type: none"> <li>• <code>expand</code> Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation.</li> <li>• <code>LE</code> Applies the Leading Eigenvalue algorithm (<code>cluster_leading_eigen</code>) on the empirical correlation matrix. If the number of dimensions is 1, then the Leading Eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation.</li> <li>• <code>louvain</code> Applies the Louvain algorithm (<code>cluster_louvain</code>) on the empirical correlation matrix using a resolution parameter = 0.95. If the number of dimensions is 1, then the Louvain solution is used; otherwise, regular EGA is used. This method was validated in the Christensen (2022) simulation.</li> </ul> |
| <code>model</code>            | <p>Character. A string indicating the method to use. Current options are:</p> <ul style="list-style-type: none"> <li>• <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</li> <li>• <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>model.args</code>       | List. A list of additional arguments for <code>EBICglasso.qgraph</code> or <code>TMFG</code> . By default, <code>gamma</code> is set to 0 in <code>EBICglasso.qgraph</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>algorithm</code>        | <p>A string indicating the algorithm to use or a function from <code>igraph</code> Current options are:</p> <ul style="list-style-type: none"> <li>• <code>walktrap</code> Computes the Walktrap algorithm using <code>cluster_walktrap</code></li> <li>• <code>leiden</code> Computes the Leiden algorithm using <code>cluster_leiden</code></li> <li>• <code>louvain</code> Computes the Louvain algorithm using <code>cluster_louvain</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>algorithm.args</code>   | List. A list of additional arguments for <code>cluster_walktrap</code> , <code>cluster_louvain</code> , or some other community detection algorithm function (see examples)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>consensus.method</code> | <p>Character. What consensus clustering method should be used? Defaults to "highest_modularity". Current options are:</p> <ul style="list-style-type: none"> <li>• <code>highest_modularity</code> Uses the community solution that achieves the highest modularity across iterations</li> <li>• <code>most_common</code> Uses the community solution that is found the most across iterations</li> <li>• <code>iterative</code> Identifies the most common community solutions across iterations and determines how often nodes appear in the same community together. A threshold of 0.30 is used to set low proportions to zero. This process repeats iteratively until all nodes have a proportion of 1 in the community solution.</li> <li>• <code>lowest_tefi</code> Uses the community solution that achieves the lowest <code>tefi</code> across iterations</li> </ul>                                                                                                                                                                                                                                                                                                                      |

|                             |                                                                                                                                                                                                                                                                                                                        |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>consensus.iter</code> | Numeric. Number of iterations to perform in consensus clustering for the Louvain algorithm (see Lancichinetti & Fortunato, 2012). Defaults to 100                                                                                                                                                                      |
| <code>ncores</code>         | 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<br>If you're unsure how many cores your computer has, then use the following code: <code>parallel::detectCores()</code> |
| <code>progress</code>       | Boolean. Should progress be displayed? Defaults to TRUE. For Windows, FALSE is about 2x faster                                                                                                                                                                                                                         |

### Value

Returns a list containing:

|                          |                                                                                                                                                                                                                                                                                                                |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>memberships</code> | Original memberships provided in <code>memberships</code> or from <a href="#">EGA</a> if NULL                                                                                                                                                                                                                  |
| <code>EGA</code>         | Original <a href="#">EGA</a> results for the sample                                                                                                                                                                                                                                                            |
| <code>groups</code>      | <ul style="list-style-type: none"> <li>• <a href="#">EGA</a> EGA results for each group</li> <li>• <code>loadings</code> Network loadings for each group</li> <li>• <code>loadingsDifference</code> Difference between the dominant loadings of each group</li> </ul>                                          |
| <code>permutation</code> | <ul style="list-style-type: none"> <li>• <code>groups</code> Permuted groups across iterations</li> <li>• <code>loadings</code> Loadings for each group for each permutation</li> <li>• <code>loadingsDifference</code> Difference between the dominant loadings of each group for each permutation</li> </ul> |
| <code>results</code>     | Data frame of the results (which are printed)                                                                                                                                                                                                                                                                  |

### Author(s)

Laura Jamison <lj5yn@virginia.edu>, Alexander P. Christensen <alexpaulchristensen@gmail.com>, and Hudson F. Golino <hfg9s@virginia.edu>

### Examples

```
Load data
wmt <- wmt2[-1,7:24]

Groups
groups <- rep(1:2, each = nrow(wmt) / 2)

Not run:
Measurement invariance
results <- invariance(wmt, groups, ncores = 2)
End(Not run)
```

---

|                            |                                                            |
|----------------------------|------------------------------------------------------------|
| <code>itemStability</code> | <i>Item Stability Statistics from <code>bootEGA</code></i> |
|----------------------------|------------------------------------------------------------|

---

### 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, IS.plot = TRUE, structure = NULL, ...)
```

### Arguments

|                          |                                                                                                                                                                      |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>bootega.obj</code> | A <code>bootEGA</code> object                                                                                                                                        |
| <code>IS.plot</code>     | Should the plot be produced for <code>item.replication</code> ? If TRUE, then a plot for the <code>item.replication</code> output will be produced. Defaults to TRUE |
| <code>structure</code>   | User specified dimensionality structure.                                                                                                                             |
| <code>...</code>         | Additional arguments. Used for deprecated arguments from previous versions of <code>itemStability</code>                                                             |

### Value

Returns a list containing:

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>membership</code>     | A list containing: <ul style="list-style-type: none"> <li>• <code>empirical</code> The empirical memberships from the empirical <code>EGA</code> result</li> <li>• <code>unique</code> The unique dimensions from the empirical <code>EGA</code> result</li> <li>• <code>bootstrap</code> The memberships from the replicate samples in the <code>bootEGA</code> results</li> </ul>                                                                                                                                                  |
| <code>item.stability</code> | A list containing: <ul style="list-style-type: none"> <li>• <code>empirical.dimensions</code> The proportion of times each item replicated within the empirical <code>EGA</code> defined dimension. This <code>EGA</code> result is defined using the input from <code>bootEGA</code></li> <li>• <code>all.dimensions</code> The proportion of times each item replicated in each of the empirical <code>EGA</code> defined dimensions. This <code>EGA</code> result is defined using the input from <code>bootEGA</code></li> </ul> |
| <code>plot</code>           | A plot of the number of times each item replicated within the empirical <code>EGA</code> defined dimension.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>mean.loadings</code>  | Matrix of the average standardized network loading (computed using <code>net.loads</code> ) for each item in each dimension                                                                                                                                                                                                                                                                                                                                                                                                          |

**Author(s)**

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

**References**

Christensen, A. P., & Golino, H. (2021). Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis: A tutorial. *Psych*, 3(3), 479-500.

Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, 34(6), 1095-1108.

**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:
Standard EGA example
boot.wmt <- bootEGA(
 data = wmt, iter = 100, # recommended 500
 plot.typicalStructure = FALSE, # No plot for CRAN checks
 type = "parametric", ncores = 2
)

Standard item stability
wmt.is <- itemStability(boot.wmt)

Produce Methods section
methods.section(
 boot.wmt,
 stats = "itemStability"
)

EGA fit example
boot.wmt.fit <- bootEGA(
 data = wmt, iter = 500,
 EGA.type = "EGA.fit",
 type = "parametric", ncores = 2
)

EGA fit item stability
wmt.is.fit <- itemStability(boot.wmt.fit)

Hierarchical EGA example
boot.wmt.hier <- bootEGA(
 data = wmt, iter = 500,
```

```

 EGA.type = "hierEGA",
 type = "parametric", ncores = 2
)

Hierarchical EGA item stability
wmt.is.hier <- itemStability(boot.wmt.hier)

Random-intercept EGA example
boot.wmt.ri <- bootEGA(
 data = wmt, iter = 500,
 EGA.type = "riEGA",
 type = "parametric", ncores = 2
)

Random-intercept EGA item stability
wmt.is.ri <- itemStability(boot.wmt.ri)
End(Not run)

```

---

 jsd

*Jensen-Shannon Distance*


---

### Description

Computes the Jensen-Shannon Distance between two networks

### Usage

```
jsd(network1, network2, method = c("kld", "spectral"))
```

### Arguments

|          |                                                                                                                                                                                                                                                                                              |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| network1 | Matrix or data frame. Network to be compared                                                                                                                                                                                                                                                 |
| network2 | Matrix or data frame. Second network to be compared                                                                                                                                                                                                                                          |
| method   | Character. Method to compute Jensen-Shannon Distance. Defaults to "spectral".<br>Options: <ul style="list-style-type: none"> <li>• "kld" Uses Kullback-Leibler Divergence</li> <li>• "spectral" Uses eigenvalues of combinatorial Laplacian matrix to compute Von Neumann entropy</li> </ul> |

### Value

Returns Jensen-Shannon Distance

### Author(s)

Hudson Golino <hfg9s at virginia.edu> & Alexander P. Christensen <alexander.christensen at Vanderbilt.Edu>

**Examples**

```

Obtain wmt2 data
wmt <- wmt2[,7:24]

Set seed (for reproducibility)
set.seed(1234)

Split data
split1 <- sample(
 1:nrow(wmt), floor(nrow(wmt) / 2)
)
split2 <- setdiff(1:nrow(wmt), split1)

Obtain split data
data1 <- wmt[split1,]
data2 <- wmt[split2,]

Perform EBICglasso
glas1 <- EBICglasso.qgraph(data1)
glas2 <- EBICglasso.qgraph(data2)

Spectral JSD
jsd(glas1, glas2) # 0.1618195

Spectral JSS (similarity)
1 - jsd(glas1, glas2) # 0.8381805

Jensen-Shannon Divergence
jsd(glas1, glas2, method = "kld") # 0.1923636

```

---

jsd.ergoInfo

*Jensen-Shannon Distance Test for Ergodicity*


---

**Description**

Tests the Jensen-Shannon Distance ([jsd](#)) of each individual's network structure to the population's network structure. Using a threshold, the proportion of [jsd](#) values **greater** than the threshold are computed. If this proportion is **greater** or equal to the maximum allowable proportion of individuals, then the system is determined to be nonergodic. If the proportion is **fewer** than the maximum allowable proportion, then the system is determined to be ergodic

**Usage**

```

jsd.ergoInfo(
 dynEGA.object,
 method = c("kld", "spectral"),
 threshold = 0.2,
 max.proportion = 0.1
)

```



**Arguments**

|                |                                                                                                                                                                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dynEGA.object  | A <a href="#">dynEGA</a> or a <a href="#">dynEGA.ind.pop</a> object that is used to match the arguments of the EII object.                                                                                                                                                                   |
| method         | Character. Method to compute Jensen-Shannon Distance. Defaults to "spectral".<br>Options: <ul style="list-style-type: none"> <li>• "kld" Uses Kullback-Leibler Divergence</li> <li>• "spectral" Uses eigenvalues of combinatorial Laplacian matrix to compute Von Neumann entropy</li> </ul> |
| threshold      | Numeric. Sets the threshold of the <a href="#">jsd</a> determined to be <b>too distant</b> for ergodicity to hold. Defaults to 0.20. Values can range between 0 and 1                                                                                                                        |
| max.proportion | Numeric. Sets the proportion of <a href="#">jsd</a> values that are <b>greater</b> than the specified threshold. Defaults to 0.10. Values can range between 0 and 1                                                                                                                          |

**Value**

Returns a list containing:

|                |                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| boot.ergoInfo  | The values of the Ergodicity Information Index obtained in the bootstrap                                                                                                                                                                              |
| p.value        | The two-sided *p*-value of the bootstrap test for the Ergodicity Information Index. The null hypothesis is that the empirical Ergodicity Information index is equal to the expected value of the EII with small variation in the population structure |
| effect         | Indicates whether the empirical EII is greater or less than the bootstrap distribution of EII.                                                                                                                                                        |
| interpretation | How you can interpret the result of the test in plain English                                                                                                                                                                                         |
| plot.dist      | Histogram of the bootstrapped ergodicity information index                                                                                                                                                                                            |
| methods        | Methods to report for print/summary S3methods and automated Methods section                                                                                                                                                                           |

**Author(s)**

Hudson Golino <hfg9s at virginia.edu> & Alexander P. Christensen <alexander.christensen at Vanderbilt.Edu>

**References**

Golino, H., Nesselroade, J., & Christensen, A. P. (2022). Toward a psychology of individuals: The ergodicity information index and a bottom-up approach for finding generalizations. *PsyArXiv*.

**Examples**

```
Not run:
Dynamic EGA individual and population structures
dyn1 <- dynEGA.ind.pop(
 data = sim.dynEGA[,-26], n.embed = 5, tau = 1,
 delta = 1, id = 25, use.derivatives = 1,
```

```

 model = "glasso", ncores = 2, corr = "pearson"
)

JSD Ergodicity Test
testing.ergoInfo <- jsd.ergoInfo(
 dynEGA.object = dyn1
)
End(Not run)

```

---

LCT

*Loadings Comparison Test*


---

### Description

An algorithm to identify whether data were generated from a 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,
 dynamic = FALSE,
 dynamic.args = list(n.embed = 4, tau = 1, delta = 1, use.derivatives = 1)
)

```

### Arguments

|                           |                                                                                                                                                                                                                                                                                                 |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>data</code>         | Matrix or data frame. A data frame 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                                                                                                                                                 |
| <code>dynamic</code>      | Boolean. Is the dataset a time series where rows are time points and columns are variables? Defaults to <code>FALSE</code> .                                                                                                                                                                    |
| <code>dynamic.args</code> | List. Arguments to be used in <code>dynEGA</code> . Defaults: <ul style="list-style-type: none"> <li>• <code>n.embed</code> Number of embeddings: 4</li> <li>• <code>tau</code> Lag: 1</li> <li>• <code>delta</code> Delta: 1</li> <li>• <code>use.derivatives</code> Derivatives: 1</li> </ul> |

**Value**

Returns a list containing:

|            |                                                                                           |
|------------|-------------------------------------------------------------------------------------------|
| empirical  | Prediction of model based on empirical dataset only                                       |
| bootstrap  | Prediction of model based on means of the loadings across the bootstrap replicate samples |
| proportion | Proportions of models suggested across bootstraps                                         |

**Author(s)**

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

**References**

Christensen, A. P., & Golino, H. (2021). Factor or network model? Predictions from neural networks. *Journal of Behavioral Data Science*, 1(1), 85-126.

**Examples**

```
Compute LCT
Factor model
LCT(data = psychTools::bfi[,1:25])

Not run:
Dynamic LCT
LCT(sim.dynEGA[sim.dynEGA$ID == 1,1:24], dynamic = TRUE)
End(Not run)
```

---

louvain

*Louvain Community Detection Algorithm*

---

**Description**

Computes the Louvain community detection algorithm (Blondel et al., 2008)

**Usage**

```
louvain(A, method = c("modularity", "tefi"), resolution = 1, corr = NULL)
```

**Arguments**

|            |                                                                                                                                                                       |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A          | Matrix or data frame. A network adjacency matrix                                                                                                                      |
| method     | Character. Whether modularity or <code>tefi</code> should be used to optimize communities. Defaults to "modularity"                                                   |
| resolution | Numeric. Resolution parameter for computing modularity. Defaults to 1. Values smaller than 1 favor larger communities; values larger than 1 favor smaller communities |
| corr       | Matrix or data frame. Correlation matrix to be used when method = "tefi"                                                                                              |

**Details**

This version was adapted from the Matlab code available here: <https://perso.uclouvain.be/vincent.blondel/research/louvain.htm>. The code was adjusted to mirror the results of `cluster_louvain`. The Louvain algorithm's results can vary depending on node ordering. In this version, nodes are **not** shuffled so that consistent results can be achieved with the same node ordering. Results from `cluster_louvain` will shuffle nodes **within** the function and therefore will sometimes produce similar results and sometimes produce slightly different results. This version is based all in R and therefore is slower than the version in `igraph`.

**Value**

Returns a list containing:

|            |                                                                                |
|------------|--------------------------------------------------------------------------------|
| wc         | A matrix of lower to higher order community membership detected in the network |
| modularity | A vector of modularity values corresponding the rows of the wc matrix          |

**Author(s)**

Alexander P. Christensen <[alexpaulchristensen@gmail.com](mailto:alexpaulchristensen@gmail.com)> and Hudson Golino <[hfg9s@virginia.edu](mailto:hfg9s@virginia.edu)>

**References**

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008, P10008.

**Examples**

```
Load data
dep <- depression[,24:44]

Estimate correlations
corr <- qgraph::cor_auto(dep)

Estimate network
net <- EBICglasso.qgraph(corr, n = nrow(dep))

Estimate communities using modularity
louvain(net, method = "modularity")
```

```
Estimate communities using tefi
louvain(net, method = "tefi", corr = corr)
```

---

mctest.ergoInfo

*Monte-Carlo Test for the Ergodicity Information Index*


---

## Description

Computes a Monte-Carlo Test for the Ergodicity Information Index, comparing the empirical Ergodicity Information index to values obtained in a Monte-Carlo simulation in which all individuals have a similar latent structure. The p-values in the Monte-Carlo test can be calculated as  $(\text{sum}(\text{EII} \geq \text{MC.EII}) + 1) / (\text{iter} + 1)$  and as  $(\text{sum}(\text{EII} \leq \text{MC.EII}) + 1) / (\text{iter} + 1)$ , where EII is the empirical Ergodicity Information Index, MC.EII is the values of the Ergodicity Information Index obtained in the simulation, and *iter* is the number of random samples generated in the simulation. The two-sided p-value is computed as two times the lowest p-value. In the Monte-Carlo Test for the Ergodicity Information Index, the null hypothesis is that the empirical value of EII is equal to the Monte-Carlo value of EII obtained in multiple individuals with a similar latent structure. Small values of p indicate that is very unlikely to obtain an EII as large as the one obtained in the empirical sample if the null hypothesis is true, thus there is convincing evidence that the empirical Ergodicity Information Index is different than it could be expected if all individuals had a similar latent structure, conditioned on the parameters used to simulate the data.

## Usage

```
mctest.ergoInfo(
 iter,
 N,
 EII,
 use,
 variab,
 timep,
 nfact,
 error,
 dfm,
 loadings,
 autoreg,
 crossreg,
 var.shock,
 cov.shock,
 embed,
 tau,
 delta,
 derivatives,
 model,
 model.args = list(),
```

```

algorithm = c("walktrap", "louvain"),
algorithm.args = list(),
corr,
ncores,
...
)

```

### Arguments

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| iter      | Numeric integer. Number of random samples to generate in the Monte-Carlo simulation. At least 500 is recommended                                                                                                                                                                                                                                                                                                                                                        |
| N         | Numeric integer. Number of individuals to simulate data from, using the <a href="#">simDFM</a> function.                                                                                                                                                                                                                                                                                                                                                                |
| EII       | Numeric. Empirical Ergodicity Information Index obtained via the <a href="#">ergoInfo</a> function.                                                                                                                                                                                                                                                                                                                                                                     |
| use       | Character. A string indicating what network element will be used to compute the algorithm complexity in the <a href="#">ergoInfo</a> function, the list of edges or the weights of the network. Defaults to use = "edge.list". Current options are: <ul style="list-style-type: none"> <li>• edge.list Calculates the algorithm complexity using the list of edges.</li> <li>• weights Calculates the algorithm complexity using the weights of the network.</li> </ul> |
| 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. Defaults to "DAFS". Current options are: <ul style="list-style-type: none"> <li>• DAFS Simulates data using the direct autoregressive factor score model. This is the default method</li> <li>• RandomWalk Simulates data using a dynamic factor model with random walk factor scores.</li> </ul>                                                                                                                |
| loadings  | Magnitude of the loadings.                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| autoreg   | Magnitude of the autoregression coefficients. Default is "autoreg = 0.8".                                                                                                                                                                                                                                                                                                                                                                                               |
| crossreg  | Magnitude of the cross-regression coefficients. Default is "crossreg = 0.1".                                                                                                                                                                                                                                                                                                                                                                                            |
| var.shock | Magnitude of the random shock variance. Default is "var.shock = 0.18".                                                                                                                                                                                                                                                                                                                                                                                                  |
| cov.shock | Magnitude of the random shock covariance Default is "cov.shock = 0.36".                                                                                                                                                                                                                                                                                                                                                                                                 |
| embed     | Integer. Number of embedded dimensions (the number of observations to be used in the <a href="#">Embed</a> function). For example, an "embed = 5" will use five observations to estimate a single derivative. Defaults to embed = 5.                                                                                                                                                                                                                                    |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tau            | Integer. Number of observations to offset successive embeddings in the <a href="#">Embed</a> 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".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| derivatives    | Integer. The order of the derivative to be used in the EGA procedure. Default to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| model          | Character. A string indicating the method to use. Defaults to <code>glasso</code> . Current options are: <ul style="list-style-type: none"> <li>• <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</li> <li>• <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph</li> </ul>                                                                                                                                                        |
| model.args     | List. A list of additional arguments for <a href="#">EBICglasso.qgraph</a> or <code>TMFG</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| algorithm      | A string indicating the algorithm to use or a function from <a href="#">igraph</a><br>Current options are: <ul style="list-style-type: none"> <li>• <code>walktrap</code> Computes the Walktrap algorithm using <a href="#">cluster_walktrap</a></li> <li>• <code>louvain</code> Computes the Walktrap algorithm using <a href="#">cluster_louvain</a></li> </ul>                                                                                                                                                                                                                                       |
| algorithm.args | List. A list of additional arguments for <a href="#">cluster_walktrap</a> , <a href="#">cluster_louvain</a> , or some other community detection algorithm function (see examples)                                                                                                                                                                                                                                                                                                                                                                                                                       |
| corr           | Type of correlation matrix to compute. The default uses <code>cor_auto</code> . Current options are: <ul style="list-style-type: none"> <li>• <code>cor_auto</code> Computes the correlation matrix using the <a href="#">cor_auto</a> function from <a href="#">qgraph</a>.</li> <li>• <code>pearson</code> Computes Pearson's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> <li>• <code>spearman</code> Computes Spearman's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> </ul> |
| 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<br>If you're unsure how many cores your computer has, then use the following code: <code>parallel::detectCores()</code>                                                                                                                                                                                                                            |
| ...            | Additional arguments. Used for deprecated arguments from previous versions of <a href="#">EGA</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

## Value

Returns a list containing:

|                  |                                                                                                                                                                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mc.ergoInfo      | The values of the Ergodicity Information Index obtained in the Monte-Carlo Simulation                                                                                                                                                            |
| p.value.twosided | The p-value of the Monte-Carlo test for the Ergodicity Information Index. The null hypothesis is that the empirical Ergodicity Information index is equal to the expected value of the EII if the all individuals had similar latent structures. |

|           |                                                                                          |
|-----------|------------------------------------------------------------------------------------------|
| effect    | Indicates wheter the empirical EII is greater or less then the Monte-Carlo obtained EII. |
| plot.dist | Histogram of the bootstrapped ergodicity information index                               |

**Author(s)**

Hudson Golino <hfg9s at virginia.edu>

**Examples**

```
Not run:
\donttest{
dyn1 <- dynEGA.ind.pop(data = sim.dynEGA, n.embed = 5, tau = 1,
 delta = 1, id = 21, group = 22, use.derivatives = 1,
 model = "glasso", ncores = 2, corr = "pearson")

eii1 <- ergoInfo(data = dyn1)$EII

dist.ergoInfo <- mctest.ergoInfo(iter = 10, N = 10, EII = eii1,
 variab = 4,
 timep = 100, nfact = 2, error = 0.05, dfm = "DAFS", loadings = 0.55, autoreg = 0.8,
 crossreg = 0.1, var.shock = 0.18, cov.shock = 0.36, embed = 5, tau=1, delta=1, derivatives=1,
 model = "glasso", ncores = 2, corr = "pearson")
}
End(Not run)
```

---

methods.section

*Automated Methods Section for [EGAnet](#) Objects*

---

**Description**

This function accepts [EGA](#) objects and generates a Methods section for your analysis. The output is an HTML page containing the descriptions of the methods and parameters as well as a Reference section for appropriate citation.

**Usage**

```
methods.section(
 ...,
 stats = c("net.loads", "net.scores", "dimensionStability", "itemStability")
)
```



**Arguments**

- ... [EGAnet](#) objects. Available methods (more methods will be added soon!):
- [EGA](#) Exploratory graph analysis
  - [bootEGA](#) Bootstrap exploratory graph analysis
  - [UVA](#) Unique variable analysis
- stats Methods section for statistics in [EGAnet](#). Multiple statistics can be input. Available statistics:
- [net.loads](#) Network loadings. Requires [EGA](#) object to be input
  - [net.scores](#) Network scores. Requires [EGA](#) object to be input
  - [dimensionStability](#) Structural consistency. Requires [bootEGA](#) object to be input
  - [itemStability](#) Item stability. Requires [bootEGA](#) object to be input

**Value**

Automated HTML Methods section in your default browser

**Examples**

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

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

EGA Methods section
methods.section(ega.wmt)

Estimate standardized network loadings
wmt.loads <- net.loads(ega.wmt)$std

EGA Methods section with network loadings
methods.section(ega.wmt, stats = "net.loads")

bootEGA example
plot.type = "qqgraph" used for CRAN checks
plot.type = "GGally" is the default
boot.wmt <- bootEGA(
 data = wmt, iter = 500,
 type = "parametric", ncores = 2
)

EGA and bootEGA Methods section
methods.section(ega.wmt, boot.wmt)

Estimate structural consistency
sc.wmt <- dimensionStability(boot.wmt)
```

```

EGA and bootEGA Methods section with structural consistency and item stability
methods.section(boot.wmt, stats = c("dimensionStability", "itemStability"))

EGA with network loadings and
bootEGA Methods section with structural consistency and item stability
methods.section(
 ega.wmt,
 boot.wmt,
 stats = c(
 "net.loads",
 "dimensionStability",
 "itemStability"
)
)
End(Not run)

```

---

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)
```

### Arguments

|              |                                                                                                                                                                                                                                                                                                               |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A            | Matrix, data frame, or <a href="#">EGA</a> object. A network adjacency matrix                                                                                                                                                                                                                                 |
| wc           | Numeric or character vector. A vector of community assignments. If input into A is an <a href="#">EGA</a> 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 |

### 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:

|                      |                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>unstd</code>   | A matrix of the unstandardized within- and between-community strength values for each node                               |
| <code>std</code>     | A matrix of the standardized within- and between-community strength values for each node                                 |
| <code>minLoad</code> | The minimum loading to appear in summary of network loadings. Use <code>print()</code> or <code>summary()</code> to view |

## Author(s)

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

## References

Christensen, A. P., & Golino, H. (2021). On the equivalency of factor and network loadings. *Behavior Research Methods*, *53*, 1563-1580.

Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, *34*, 1095-1108.

Hallquist, M., Wright, A. C. G., & Molenaar, P. C. M. (2019). Problems with centrality measures in psychopathology symptom networks: Why network psychometrics cannot escape psychometric theory. *Multivariate Behavioral Research*, 1-25.

## Examples

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

Not run:
Estimate EGA
ega.wmt <- EGA(
 data = wmt,
 plot.EGA = FALSE # No plot for CRAN checks
)
End(Not run)

Network loadings
net.loads(ega.wmt)
```

```
Not run:
Produce Methods section
methods.section(
 ega.wmt,
 stats = "net.loads"
)
End(Not run)
```

---

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, ...)
```

### Arguments

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>data</code>   | Matrix or data frame. Must be a dataset                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>A</code>      | Matrix, data frame, or <a href="#">EGA</a> object. An adjacency matrix of network data                                                                                                                                                                                                                                                                                                                                                              |
| <code>wc</code>     | Numeric. A vector of community assignments. Not necessary if an <a href="#">EGA</a> object is input for argument <code>A</code>                                                                                                                                                                                                                                                                                                                     |
| <code>global</code> | Boolean. Should general network loadings be computed in scores? Defaults to <code>FALSE</code> . 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"> <li>• <code>none</code> No imputation is performed. This is the default.</li> <li>• <code>mean</code> The "mean" value of the columns are used to replace the missing data.</li> <li>• <code>median</code> The "median" value of the columns are used to replace the missing data.</li> </ul> |
| <code>...</code>    | Additional arguments for <a href="#">EGA</a>                                                                                                                                                                                                                                                                                                                                                                                                        |

### 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 <a href="#">net.loads</a> ) |

**Author(s)**

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

**References**

Christensen, A. P., & Golino, H. (2021). On the equivalency of factor and network loadings. *Behavior Research Methods*, *53*, 1563-1580.

Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, *34*, 1095-1108.

Golino, H., Christensen, A. P., Moulder, R., Kim, S., & Boker, S. M. (2021). 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. *Psychometrika*.

**Examples**

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

Not run:
Estimate EGA
ega.wmt <- EGA(
 data = wmt,
 plot.EGA = FALSE # No plot for CRAN checks
)
End(Not run)

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

Not run:
Produce Methods section
methods.section(
 ega.wmt,
 stats = "net.scores"
)
```

```
End(Not run)
```

---

```
network.descriptives Descriptive Statistics for Networks
```

---

### Description

Computes descriptive statistics for network models

### Usage

```
network.descriptives(network)
```

### Arguments

network            Matrix, data frame, [qgraph](#), or [EGA](#) object

### Value

Numeric vector including:

|                 |                                                                                                                                                                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mean_weight     | The average of the edge weights in the network                                                                                                                                                                                                                                                                                        |
| SD_weight       | The standard deviation of the edge weights in the network                                                                                                                                                                                                                                                                             |
| Min_weight      | The minimum of the edge weights in the network                                                                                                                                                                                                                                                                                        |
| Max_weight      | The maximum of the edge weights in the network                                                                                                                                                                                                                                                                                        |
| Density         | The density of the network                                                                                                                                                                                                                                                                                                            |
| ASPL            | The average shortest path length (ASPL) of the network (computed as unweighted)                                                                                                                                                                                                                                                       |
| CC              | The clustering coefficient (CC) of the network (computed as unweighted)                                                                                                                                                                                                                                                               |
| swn.rand        | Small-worldness measure based on random networks:<br>$swn.rand = (ASPL/ASPL_{r,andom})/(CC/CC_{r,andom})$ swn.rand > 1 suggests the network is small-world                                                                                                                                                                            |
| swn.HG          | Small-worldness measure based on Humphries & Gurney (2008):<br>$swn.HG = (transitivity/transitivity_{r,andom})/(ASPL/ASPL_{r,andom})$ swn.HG > 1 suggests the network is small-world                                                                                                                                                  |
| swn.TJHBL       | Small-worldness measure based on Telesford, Joyce, Hayasaka, Burdette, & Laurienti (2011):<br>$swn.TJHBL = (ASPL_{r,andom}/ASPL) - (CC/CC_{lattice})$ swn.TJHBL near 0 suggests the network is small-world, positive values suggest more random network characteristics, negative values suggest more lattice network characteristics |
| scale-free_R-sq | The R-squared fit of whether the degree distribution follows the power-law (many small degrees, few large degrees)                                                                                                                                                                                                                    |

**Author(s)**

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

**References**

```
swn.HG
Humphries, M. D., & Gurney, K. (2008). Network 'small-world-ness': A quantitative method for
determining canonical network equivalence. PLoS one, 3, e0002051
swn.TJHBL
Telesford, Q. K., Joyce, K. E., Hayasaka, S., Burdette, J. H., & Laurienti, P. J. (2011). The ubiquity
of small-world networks. Brain Connectivity, 1(5), 367-375
scale-free_R-sq
Langfelder, P., & Horvath, S. (2008). WGCNA: an R package for weighted correlation network
analysis. BMC Bioinformatics, 9, 559
```

**Examples**

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

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

Compute descriptives
network.descriptives(ega.wmt)
```

---

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.

**Examples**

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

---

plots

*S3Methods for Plotting*


---

**Description**

Plots for EGAnet objects

**Usage**

```
S3 method for class 'bootEGA'
plot(x, title = "",
 plot.args = list(), produce = TRUE, ...)

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

S3 method for class 'dynEGA'
plot(x, title = "",
 plot.args = list(), produce = TRUE, ...)

S3 method for class 'dynEGA.Groups'
plot(x, ncol, nrow, title = "",
 plot.args = list(), produce = TRUE, ...)

S3 method for class 'dynEGA.Individuals'
plot(x, title = "", id = NULL,
 plot.args = list(), produce = TRUE, ...)

S3 method for class 'EGA'
plot(x, title = "",
 plot.args = list(), produce = TRUE, ...)
```

**Arguments**

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>         | Object from EGAnet package                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>title</code>     | Character. Title of the plot. Defaults to ""                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>plot.args</code> | List. A list of additional arguments for the network plot. See <a href="#">ggnet2</a> for full list of arguments: <ul style="list-style-type: none"> <li>• <code>vsize</code> Size of the nodes. Defaults to 6.</li> <li>• <code>label.size</code> Size of the labels. Defaults to 5.</li> <li>• <code>alpha</code> The level of transparency of the nodes, which might be a single value or a vector of values. Defaults to 0.7.</li> </ul> |



|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | <ul style="list-style-type: none"> <li>• <code>edge.alpha</code> The level of transparency of the edges, which might be a single value or a vector of values. Defaults to 0.4.</li> <li>• <code>legend.names</code> A vector with names for each dimension</li> <li>• <code>color.palette</code> The color palette for the nodes. For custom colors, enter HEX codes for each dimension in a vector. See <a href="#">color_palette_EGA</a> for more details and examples</li> </ul> |
| <code>produce</code> | Boolean. This argument is used internally. Should plot be produced? Defaults to TRUE                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>...</code>     | Arguments passed on to <ul style="list-style-type: none"> <li>• <a href="#">semPaths</a> Functions: CFA</li> </ul>                                                                                                                                                                                                                                                                                                                                                                  |
| <code>vsize</code>   | Numeric. Size of vertices in <a href="#">CFA</a> plots. Defaults to 6                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>layout</code>  | Character. Layout of plot (see <a href="#">semPaths</a> ). Defaults to "spring"                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>ncol</code>    | Numeric. Number of columns                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>nrow</code>    | Numeric. Number of rows                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>id</code>      | 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>

---

|           |                                      |
|-----------|--------------------------------------|
| prime.num | <i>Prime Numbers through 100,000</i> |
|-----------|--------------------------------------|

---

**Description**

Numeric vector of primes generated from the primes package. Used in the function `[EGAnet]{ergoInfo}`.  
Not for general use

**Usage**

```
data(prime.num)
```

**Format**

A 1185x24 response matrix

**Examples**

```
data("prime.num")
```

---

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, ...)

S3 method for class 'invariance'
print(x, ...)

S3 method for class 'hierEGA'
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 | <i>Residualized</i> <a href="#">EGA</a> |
|-------------|-----------------------------------------|

---

### 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)
```

### 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.                                                                         |

### 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 ggmlfit function of the <a href="#">qgraph</a> 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

```
Load data
data <- optimism

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

Fit:
opt.res$Fit
End(Not run)
```

---

riEGA

*Random-Intercept EGA*


---

## Description

Estimates the number of substantive dimensions after controlling for wording effects. EGA is applied to a residual correlation matrix after subtracting and random intercept factor with equal unstandardized loadings from all the regular and unrecoded reversed items in the database

## Usage

```
riEGA(
 data,
 n = NULL,
 uni.method = c("expand", "LE", "louvain"),
 corr = c("cor_auto", "pearson", "spearman"),
 model = c("glasso", "TMFG"),
 model.args = list(),
 algorithm = c("walktrap", "louvain"),
 algorithm.args = list(),
 consensus.iter = 100,
 consensus.method = c("highest_modularity", "most_common", "iterative", "lowest_tefi"),
 plot.EGA = TRUE,
 plot.args = list(),
 estimator = c("auto", "WLSMV", "MLR"),
 lavaan.args = list()
)
```

## Arguments

**data** Matrix or data frame. Variables (down columns) or correlation matrix. If the input is a correlation matrix, then argument **n** (number of cases) is **required**. Variables **MUST** be unrecoded – reversed items should **remain** reversed

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| n                | Integer. Sample size if data provided is a correlation matrix                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| uni.method       | Character. What unidimensionality method should be used? Defaults to "louvain". Current options are: <ul style="list-style-type: none"> <li>• expand Expands the correlation matrix with four variables correlated .50. If number of dimension returns 2 or less in check, then the data are unidimensional; otherwise, regular EGA with no matrix expansion is used. This is the method used in the Golino et al. (2020) <i>Psychological Methods</i> simulation.</li> <li>• LE Applies the Leading Eigenvalue algorithm (<code>cluster_leading_eigen</code>) on the empirical correlation matrix. If the number of dimensions is 1, then the Leading Eigenvalue solution is used; otherwise, regular EGA is used. This is the final method used in the Christensen, Garrido, and Golino (2021) simulation.</li> <li>• louvain Applies the Louvain algorithm (<code>cluster_louvain</code>) on the empirical correlation matrix using a resolution parameter = 0.95. If the number of dimensions is 1, then the Louvain solution is used; otherwise, regular EGA is used. This method was validated in the Christensen (2022) simulation.</li> </ul> |
| corr             | Type of correlation matrix to compute. The default uses <code>cor_auto</code> . Current options are: <ul style="list-style-type: none"> <li>• cor_auto Computes the correlation matrix using the <code>cor_auto</code> function from <code>qgraph</code>.</li> <li>• pearson Computes Pearson's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> <li>• spearman Computes Spearman's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| model            | Character. A string indicating the method to use. Defaults to "glasso". Current options are: <ul style="list-style-type: none"> <li>• glasso Estimates the Gaussian graphical model using graphical LASSO with extended Bayesian information criterion to select optimal regularization parameter</li> <li>• TMFG Estimates a Triangulated Maximally Filtered Graph</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| model.args       | List. A list of additional arguments for <code>EBICglasso.qgraph</code> or TMFG                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| algorithm        | A string indicating the algorithm to use or a function from <code>igraph</code> Defaults to "walktrap". Current options are: <ul style="list-style-type: none"> <li>• walktrap Computes the Walktrap algorithm using <code>cluster_walktrap</code></li> <li>• louvain Computes the Louvain algorithm using <code>cluster_louvain</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| algorithm.args   | List. A list of additional arguments for <code>cluster_walktrap</code> , <code>cluster_louvain</code> , or some other community detection algorithm function (see examples)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| consensus.iter   | Numeric. Number of iterations to perform in consensus clustering for the Louvain algorithm (see Lancichinetti & Fortunato, 2012). Defaults to 100                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| consensus.method | Character. What consensus clustering method should be used? Defaults to "highest_modularity". Current options are:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

- `highest_modularity` Uses the community solution that achieves the highest modularity across iterations
- `most_common` Uses the community solution that is found the most across iterations
- `iterative` Identifies the most common community solutions across iterations and determines how often nodes appear in the same community together. A threshold of 0.30 is used to set low proportions to zero. This process repeats iteratively until all nodes have a proportion of 1 in the community solution.
- `lowest_tefi` Uses the community solution that achieves the lowest `tefi` across iterations

`plot.EGA` Boolean. If TRUE, returns a plot of the network and its estimated dimensions. Defaults to TRUE

`plot.args` List. A list of additional arguments for the network plot. For `plot.type = "qgraph"`:

- `vsize` Size of the nodes. Defaults to 6.

For `plot.type = "GGally"` (see [ggnet2](#) for full list of arguments):

- `vsize` Size of the nodes. Defaults to 6.
- `label.size` Size of the labels. Defaults to 5.
- `alpha` The level of transparency of the nodes, which might be a single value or a vector of values. Defaults to 0.7.
- `edge.alpha` The level of transparency of the edges, which might be a single value or a vector of values. Defaults to 0.4.
- `legend.names` A vector with names for each dimension
- `color.palette` The color palette for the nodes. For custom colors, enter HEX codes for each dimension in a vector. See [color\\_palette\\_EGA](#) for more details and examples

`estimator` Character. Estimator to use for random-intercept model (see [Estimators](#) for more details). Defaults to "auto", which selects "MLR" for continuous data and "WLSMV" for mixed and categorical data. Data are considered continuous data if they have 6 or more categories (see Rhemtulla, Brosseau-Liard, & Savalei, 2012)

`lavaan.args` List. If `reduce.method = "latent"`, then [lavaan](#)'s `cfa` function will be used to create latent variables to reduce variables. Arguments should be input as a list. Some example arguments (see [lavOptions](#) for full details)

## Value

Returns a list containing:

|     |                                                                                                                                                                                                                                       |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EGA | Results from <a href="#">EGA</a>                                                                                                                                                                                                      |
| RI  | A list containing information about the random-intercept model (if the model converged): <ul style="list-style-type: none"> <li>• <code>fit</code> The fit object for the random-intercept model using <a href="#">cfa</a></li> </ul> |

- `lavaan.args` The arguments used in `cfa`
- `loadings` Standardized loadings from the random-intercept model
- `correlation` Residual correlations after accounting for the random-intercept model

### Author(s)

Alejandro Garcia-Pardina <alejandrogp97@gmail.com>, Francisco J. Abad <fjose.abad@uam.es>, Alexander P. Christensen <alexpaulchristensen@gmail.com>, Hudson Golino <hfg9s@virginia.edu>, Luis Eduardo Garrido <luisgarrido@pucmm.edu.do>, and Robert Moulder <rgm4fd@virginia.edu>

### References

# Selection of CFA Estimator  
 Rhemtulla, M., Brosseau-Liard, P. E., & Savalei, V. (2012). When can categorical variables be treated as continuous? A comparison of robust continuous and categorical SEM estimation methods under suboptimal conditions. *Psychological Methods*, *17*, 354-373.

### Examples

```
Obtain example data
data <- optimism

Not run: # riEGA example
opt.res <- riEGA(data = optimism)
End(Not run)
```

---

sim.dynEGA

*sim.dynEGA Data*

---

### Description

A simulated (multivariate time series) data with 24 variables, 100 individual observations, 50 time points per individual and 2 groups of individuals.

### Usage

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

### Format

A 5000 x 26 multivariate time series

### Details

Data were generated using the `simDFM` function with the following arguments:

```
simDFM(variab = 12, timep = 50, nfact = 2, error = 0.175, dfm = "DAFS", loadings = 0.70,
 autoreg = 0.50, crossreg = 0.10, var.shock = 0.09, cov.shock = 0.30, variation = TRUE)
```

**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; Nesselrode, 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,
 variation = FALSE
)
```

**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"> <li>• DAFS Simulates data using the direct autoregressive factor score model. This is the default method</li> <li>• RandomWalk Simulates data using a dynamic factor model with random walk factor scores.</li> </ul> |
| 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.                               |
| variation | Boolean. Whether parameters should be varied. Defaults to FALSE. Set to TRUE to add slight variation to all parameters |

### Author(s)

Hudson F. Golino <hfg9s at virginia.edu>

### References

Engle, R., & Watson, M. (1981). A one-factor multivariate time series model of metropolitan wage rates. *Journal of the American Statistical Association*, 76(376), 774-781.

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.

### Examples

```
Not run:
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)
```

---

summaries

*S3Methods for summarizing*

---

### Description

summaries for EGAnet objects

**Usage**

```
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, ...)

S3 method for class 'invariance'
summary(object, ...)

S3 method for class 'hierEGA'
summary(object, ...)

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

**Arguments**

|        |                            |
|--------|----------------------------|
| object | Object from EGAnet package |
| ...    | Additional arguments       |

**Value**

summarys EGAnet object

**Author(s)**

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

---

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 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 <a href="#">EGA</a> |

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

**References**

Golino, H., Moulder, R. G., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Nesselroade, J., Sadana, R., Thiyagarajan, J. A., & Boker, S. M. (2020). Entropy fit indices: New fit measures for assessing the structure and dimensionality of multiple latent variables. *Multivariate Behavioral Research*.

**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)
```

**Description**

Applies the Triangulated Maximally Filtered Graph (TMFG) filtering method (**please see and cite Massara et al., 2016**). The TMFG method uses a structural constraint that limits the number of zero-order correlations included in the network ( $3n - 6$ ; where  $n$  is the number of variables). The TMFG algorithm begins by identifying four variables which have the largest sum of correlations to all other variables. Then, it iteratively adds each variable with the largest sum of three correlations to nodes already in the network until all variables have been added to the network. This structure can be associated with the inverse correlation matrix (i.e., precision matrix) to be turned into a GGM (i.e., partial correlation network) by using Local-Global Inversion Method (see Barfuss et al., 2016 for more details). See Details for more information on this network estimation method.

**Usage**

TMFG(cormat)

**Arguments**

cormat            A correlation matrix

**Details**

The TMFG method applies a structural constraint on the network, which restrains the network to retain a certain number of edges ( $3n-6$ , where  $n$  is the number of nodes; Massara et al., 2016). The network is also composed of 3- and 4-node cliques (i.e., sets of connected nodes; a triangle and tetrahedron, respectively). The TMFG method constructs a network using zero-order correlations and the resulting network can be associated with the inverse covariance matrix (yielding a GGM; Barfuss, Massara, Di Matteo, & Aste, 2016). Notably, the TMFG can use any association measure and thus does not assume the data is multivariate normal.

Construction begins by forming a tetrahedron of the four nodes that have the highest sum of correlations that are greater than the average correlation in the correlation matrix. Next, the algorithm iteratively identifies the node that maximizes its sum of correlations to a connected set of three nodes (triangles) already included in the network and then adds that node to the network. The process is completed once every node is connected in the network. In this process, the network automatically generates what's called a planar network. A planar network is a network that could be drawn on a sphere with no edges crossing (often, however, the networks are depicted with edges crossing; Tumminello, Aste, Di Matteo, & Mantegna, 2005).

**Value**

Returns a list containing:

|            |                                           |
|------------|-------------------------------------------|
| A          | The filtered adjacency matrix             |
| separators | The separators (3-cliques) in the network |
| cliques    | The cliques (4-cliques) in the network    |

**Author(s)**

Alexander Christensen <alexpaulchristensen@gmail.com>

**References**

Barfuss, W., Massara, G. P., Di Matteo, T., & Aste, T. (2016). Parsimonious modeling with information filtering networks. *Physical Review E*, *94*, 062306.

Christensen, A. P., Kenett, Y. N., Aste, T., Silvia, P. J., & Kwapił, T. R. (2018). Network structure of the Wisconsin Schizotypy Scales-Short Forms: Examining psychometric network filtering approaches. *Behavior Research Methods*, *50*, 2531-2550.

Massara, G. P., Di Matteo, T., & Aste, T. (2016). Network filtering for big data: Triangulated maximally filtered graph. *Journal of Complex Networks*, *5*, 161-178.

**Examples**

```
Pearson's correlation only for CRAN checks
A <- TMFG(cor(wmt2[,7:24]))$A
```

---

totalCor

*Total Correlation*

---

**Description**

Computes the total correlation of a dataset

**Usage**

```
totalCor(data)
```

**Arguments**

data                    Matrix or data frame. Variables to be used in the analysis

**Value**

Returns a list containing:

Ind.Entropies    Individual entropies for each variable

Joint.Entropy    The joint entropy of the dataset

Total.Cor        The total correlation of the dataset

**Author(s)**

Hudson F. Golino <hfg9s at virginia.edu>

**References**

Watanabe, S. (1960). Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development* 4, 66-82.

# Implementation Felix, L. M., Mansur-Alves, M., Teles, M., Jamison, L., & Golino, H. (2021). Longitudinal impact and effects of booster sessions in a cognitive training program for healthy older adults. *Archives of Gerontology and Geriatrics*, 94, 104337.

**Examples**

```
Compute total correlation
totalCor(wmt2[,7:24])
```

---

|             |                                 |
|-------------|---------------------------------|
| totalCorMat | <i>Total Correlation Matrix</i> |
|-------------|---------------------------------|

---

**Description**

Computes the pairwise total correlation for a dataset

**Usage**

```
totalCorMat(data)
```

**Arguments**

data                      Matrix or data frame. Variables to be used in the analysis

**Value**

Returns a square matrix with pairwise total correlations

**Author(s)**

Hudson F. Golino <hfg9s at virginia.edu>

**References**

Watanabe, S. (1960). Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development* 4, 66-82.

# Implementation Felix, L. M., Mansur-Alves, M., Teles, M., Jamison, L., & Golino, H. (2021). Longitudinal impact and effects of booster sessions in a cognitive training program for healthy older adults. *Archives of Gerontology and Geriatrics*, 94, 104337.

**Examples**

```
Not run:
Compute total correlation
totalCorMat(wmt2[,7:24])
End(Not run)
```

---

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")
```

---

UVA

*Unique Variable Analysis*

---

**Description**

Identifies redundant variables in a multivariate dataset using a number of different association methods and types of significance values (see Christensen, Garrido, & Golino, 2020 for more details)

**Usage**

```
UVA(
 data,
 n = NULL,
 model = c("glasso", "TMFG"),
 corr = c("cor_auto", "pearson", "spearman"),
 method = c("cor", "pcor", "wTO"),
 type = c("adapt", "alpha", "threshold"),
 sig,
 key = NULL,
```

```

 reduce = TRUE,
 auto = TRUE,
 label.latent = FALSE,
 reduce.method = c("latent", "remove", "sum"),
 lavaan.args = list(),
 adhoc = TRUE,
 plot.redundancy = FALSE,
 plot.args = list()
)

```

### Arguments

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data   | Matrix or data frame. Input can either be data or a correlation matrix                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| n      | Numeric. If input in data is a correlation matrix, then sample size is required. Defaults to NULL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| model  | Character. A string indicating the method to use. Current options are: <ul style="list-style-type: none"> <li>• <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</li> <li>• <code>TMFG</code> Estimates a Triangulated Maximally Filtered Graph</li> </ul>                                                                                                                                                                                    |
| corr   | Type of correlation matrix to compute. The default uses <code>cor_auto</code> . Current options are: <ul style="list-style-type: none"> <li>• <code>cor_auto</code> Computes the correlation matrix using the <code>cor_auto</code> function from <code>qgraph</code>.</li> <li>• <code>pearson</code> Computes Pearson's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> <li>• <code>spearman</code> Computes Spearman's correlation coefficient using the pairwise complete observations via the <code>cor</code> function.</li> </ul> |
| method | Character. Computes weighted topological overlap (" <code>wT0</code> " using <code>EBICglasso</code> ), partial correlations (" <code>pcor</code> "), or correlations (" <code>cor</code> ") Defaults to " <code>wT0</code> "                                                                                                                                                                                                                                                                                                                                                                     |
| type   | Character. Type of significance. Computes significance using the standard $p$ -value (" <code>alpha</code> "), adaptive alpha $p$ -value ( <code>adapt.a</code> ), or some threshold " <code>threshold</code> ". Defaults to " <code>threshold</code> "                                                                                                                                                                                                                                                                                                                                           |
| sig    | Numeric. $p$ -value for significance of overlap (defaults to <code>.05</code> ). Defaults for " <code>threshold</code> " for each method: <ul style="list-style-type: none"> <li>• "<code>wT0</code>" <code>.25</code></li> <li>• "<code>pcor</code>" <code>.35</code></li> <li>• "<code>cor</code>" <code>.50</code></li> </ul>                                                                                                                                                                                                                                                                  |
| key    | Character vector. A vector with variable descriptions that correspond to the order of variables input into data. Defaults to NULL or the column names of data                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| reduce | Boolean. Should redundancy reduction be performed? Defaults to TRUE. Set to FALSE for redundancy analysis only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |



|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>auto</code>            | Boolean. Should redundancy reduction be automated? Defaults to TRUE. Set to FALSE for manual selection                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>label.latent</code>    | Boolean. Should latent variables be labelled? Defaults to TRUE. Set to FALSE for arbitrary labelling (i.e., "LV_")                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>reduce.method</code>   | Character. How should data be reduced? Defaults to "latent" <ul style="list-style-type: none"> <li>• "latent" Redundant variables will be combined into a latent variable</li> <li>• "remove" All but one redundant variable will be removed</li> <li>• "sum" Redundant variables are combined by summing across cases (rows)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>lavaan.args</code>     | List. If <code>reduce.method = "latent"</code> , then <code>lavaan</code> 's <code>cfa</code> function will be used to create latent variables to reduce variables. Arguments should be input as a list. Some example arguments (see <code>lavOptions</code> for full details): <ul style="list-style-type: none"> <li>• <code>estimator</code> Estimator to use for latent variables (see <code>Estimators</code>) for more details. Defaults to "MLR" for continuous data and "WLSMV" for mixed and categorical data. Data are considered continuous data if they have 6 or more categories (see Rhemtulla, Brosseau-Liard, &amp; Savalei, 2012)</li> <li>• <code>missing</code> How missing data should be handled. Defaults to "fiml"</li> <li>• <code>std.lv</code> If TRUE, the metric of each latent variable is determined by fixing their (residual) variances to 1.0. If FALSE, the metric of each latent variable is determined by fixing the factor loading of the first indicator to 1.0. If there are multiple groups, <code>std.lv = TRUE</code> and "loadings" is included in the <code>group.label</code> argument, then only the latent variances <math>i</math> of the first group will be fixed to 1.0, while the latent variances of other groups are set free. Defaults to TRUE</li> </ul> |
| <code>adhoc</code>           | Boolean. Should adhoc check of redundancies be performed? Defaults to TRUE. If TRUE, adhoc check will run the redundancy analysis on the reduced variable set to determine if there are any remaining redundancies. This check is performed with the arguments: <code>method = "wTO"</code> , <code>type = "threshold"</code> , and <code>sig = .20</code> . This check is based on Christensen, Garrido, and Golino's (2020) simulation where these parameters were found to be the most conservative, demonstrating few false positives and false negatives                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>plot.redundancy</code> | Boolean. Should redundancies be plotted in a network plot? Defaults to FALSE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>plot.args</code>       | List. Arguments to be passed onto <code>ggnet2</code> . Defaults: <ul style="list-style-type: none"> <li>• <code>vsize = 6</code> Changes node size</li> <li>• <code>alpha = 0.4</code> Changes transparency</li> <li>• <code>label.size = 5</code> Changes label size</li> <li>• <code>edge.alpha = 0.7</code> Changes edge transparency</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

### Value

Returns a list:

|                         |                                                                                                                                                                                                                  |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>redundancy</code> | A list containing several objects: <ul style="list-style-type: none"> <li>• <code>redudant</code> Vectors nested within the list corresponding to redundant nodes with the name of object in the list</li> </ul> |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- data Original data
  - correlation Correlation matrix of original data
  - weights Weights determine by weighted topological overlap, partial correlation, or zero-order correlation
  - network If method = "wTO", then the network computed following [EGA](#) with [EBICglasso](#) network estimation
  - plot If redundancy.plot = TRUE, then a plot of all redundancies found
  - descriptives
    - basic A vector containing the mean, standard deviation, median, median absolute deviation (MAD), 3 times the MAD, 6 times the MAD, minimum, maximum, and critical value for the overlap measure (i.e., weighted topological overlap, partial correlation, or threshold)
    - centralTendency A matrix for all (absolute) non-zero values and their respective standard deviation from the mean and median absolute deviation from the median
  - method Returns method argument
  - type Returns type argument
  - distribution If type != "threshold", then distribution that was used to determine significance
- reduced If reduce = TRUE, then a list containing:
- data New data with redundant variables merged or removed
  - mergedA matrix containing the variables that were decided to be redundant with one another
  - methodMethod used to perform redundancy reduction
- adhoc If adhoc = TRUE, then the adhoc check containing the same objects as in the redundancy list object in the output

### Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

### References

# Simulation using UVA

Christensen, A. P., Garrido, L. E., & Golino, H. (under review). Unique Variable Analysis: A novel approach for detecting redundant variables in multivariate data. *PsyArXiv*.

# Implementation of UVA (formally node.redundant)

Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, *34*, 1095-1108.

# wTO measure

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.

```
Selection of CFA Estimator
Rhemtulla, M., Brosseau-Liard, P. E., & Savalei, V. (2012). When can categorical variables be
treated as continuous? A comparison of robust continuous and categorical SEM estimation methods
under suboptimal conditions. Psychological Methods, 17, 354-373.
```

### Examples

```
Select Five Factor Model personality items only
idx <- na.omit(match(gsub("-", ""), unlist(psychTools::spi.keys[1:5])), colnames(psychTools::spi))
items <- psychTools::spi[,idx]

Change names in redundancy output to each item's description
key.ind <- match(colnames(items), as.character(psychTools::spi.dictionary$item_id))
key <- as.character(psychTools::spi.dictionary$item[key.ind])

Not run:
Automated selection of local dependence (default)
uva.results <- UVA(data = items, key = key)

Produce Methods section
methods.section(uva.results)
End(Not run)

Manual selection of local dependence
if(interactive()){
uva.results <- UVA(data = items, key = key, auto = FALSE)}
```

---

 vn.entropy

*Entropy Fit Index using Von Neumman's entropy (Quantum Informa-  
tion 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 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 <a href="#">EGA</a> |

**Value**

Returns a list containing:

VN.Entropy.Fit The Entropy Fit Index using Von Neumann's entropy

Total.Correlation  
The total correlation of the dataset

Average.Entropy  
The average entropy of the dataset

**Author(s)**

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

**References**

Golino, H., Moulder, R. G., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Nesselroade, J., Sadana, R., Thiyagarajan, J. A., & Boker, S. M. (2020). Entropy fit indices: New fit measures for assessing the structure and dimensionality of multiple latent variables. *Multivariate Behavioral Research*.

**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**

```
Select Five Factor Model personality items only
idx <- na.omit(match(gsub("-", "", unlist(psychTools::spi.keys[1:5])), colnames(psychTools::spi)))
items <- psychTools::spi[,idx]

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

Compute entropy indices
vn.entropy(
 data = ega.spi$correlation,
 structure = ega.spi$wc
)
End(Not run)
```

---

`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

## \* datasets

- boot.wmt, 7
  - depression, 17
  - dnn.weights, 19
  - ega.wmt, 38
  - intelligenceBattery, 49
  - optimism, 71
  - prime.num, 73
  - sim.dynEGA, 79
  - toy.example, 87
  - wmt2, 93
- boot.ergoInfo, 5
- boot.wmt, 7
- bootEGA, 3, 4, 7, 7, 13, 18, 31, 35, 38, 53, 65
- CFA, 11, 12, 19, 31, 35, 38, 41, 54, 73, 75, 83, 92
- cfa, 13, 78, 79, 89
- cluster\_leading\_eigen, 8, 22, 25, 29, 36, 45, 51, 77
- cluster\_leiden, 9, 21, 25, 29, 33, 37, 46, 51
- cluster\_louvain, 8, 9, 21, 22, 25, 29, 33, 37, 44–47, 51, 60, 63, 77
- cluster\_walktrap, 9, 21, 25, 29, 33, 36, 37, 46, 51, 63, 77
- color\_palette\_EGA, 10, 14, 16, 30, 46, 73, 78
- compare.EGA.plots, 15
- convert2igraph, 17
- cor, 9, 22, 25, 29, 33, 36, 45, 46, 50, 63, 77, 88
- cor\_auto, 9, 22, 25, 29, 33, 36, 45, 50, 63, 77, 88
- depression, 17
- dimensionStability, 4, 11, 18, 65
- dnn.weights, 19
- dynEGA, 3–5, 20, 48, 57, 58
- dynEGA.ind.pop, 5, 23, 41, 48, 57
- EBICglasso, 3, 26, 88, 90
- EBICglasso.qgraph, 9, 21, 25, 26, 28, 29, 31–34, 37, 46, 51, 63, 77
- EGA, 3, 4, 7, 9–14, 18, 19, 22, 26, 28, 30, 34, 36–38, 40, 41, 44, 46, 47, 50, 52–54, 63–66, 68, 70, 75, 76, 78, 83, 90–92
- EGA.estimate, 32
- EGA.fit, 5, 9, 36
- ega.wmt, 38
- EGAnet, 15, 16, 64, 65
- EGAnet (EGAnet-package), 3
- EGAnet-package, 3
- Embed, 21, 24, 39, 43, 62, 63
- entropyFit, 3, 4, 40
- ergoInfo, 5, 41, 62
- fa, 44
- fitMeasures, 13
- ggnet2, 10, 14, 16, 30, 46, 72, 78, 89
- glasso, 26, 27
- glla, 20, 24, 43
- hclust, 48
- hierEGA, 9, 44
- igraph, 9, 17, 21, 25, 29, 30, 33, 34, 37, 46, 51, 60, 63, 77
- infoCluster, 48
- intelligenceBattery, 49
- invariance, 50
- itemStability, 4, 18, 53, 53, 65
- jsd, 55, 56, 57
- jsd.ergoInfo, 56
- lavaan, 78, 89
- lavOptions, 13, 78, 89
- LCT, 4, 58
- louvain, 59
- mctest.ergoInfo, 61

methods.section, 64  
mvrnorm, 9

net.loads, 3, 4, 53, 65, 66, 68, 69  
net.scores, 3, 44, 65, 68  
network.descriptives, 70

optimism, 71

plot.bootEGA (plots), 72  
plot.CFA (plots), 72  
plot.dynEGA (plots), 72  
plot.EGA (plots), 72  
plots, 72  
prime.num, 73  
print.dynEGA (prints), 74  
print.EGA (prints), 74  
print.hierEGA (prints), 74  
print.invariance (prints), 74  
print.NetLoads (prints), 74  
prints, 74

qgraph, 9, 14, 22, 25–27, 29, 33, 36, 45, 50,  
63, 70, 75, 77, 88

RColorBrewer, 14  
residualEGA, 75  
riEGA, 9, 76

semPaths, 13, 73  
sim.dynEGA, 79  
simDFM, 4, 62, 79, 80  
summary.dynEGA (summaries), 81  
summary.EGA (summaries), 81  
summary.hierEGA (summaries), 81  
summary.invariance (summaries), 81  
summary.NetLoads (summaries), 81  
summary.riEGA (summaries), 81  
summaries, 81

tefi, 3–5, 9, 10, 30, 34, 36, 37, 45, 47, 51, 60,  
78, 82  
TMFG, 3, 9, 21, 25, 28, 29, 31–34, 37, 51, 63, 84  
totalCor, 85  
totalCorMat, 86  
toy.example, 87

UVA, 3, 4, 65, 87

vn.entropy, 4, 91

wi2net, 27  
wmt2, 7, 38, 93