Package: psychnets (via r-universe)

June 28, 2026

Title Clean-Room Base-R Psychometric Network Estimation

Version 0.2.1

Description Estimates psychometric network models -- correlation and partial correlation, the EBIC-regularized Gaussian graphical model (graphical lasso), its nonparanormal and unregularized stepwise variants, information-filtering graphs (TMFG and LoGo), relative-importance networks, and the Ising and mixed graphical models -- reimplemented from first principles in base R with no compiled dependencies. Each regularized estimator ships a dependency-free correctness certificate (for the Gaussian graphical model, the graphical-lasso stationarity / KKT residual) so a fitted network is self-verifying: its distance from the unique optimum of its own convex objective is reported directly, rather than trusted only because it matches an external solver. The R counterpart to the 'psychaj' TypeScript library.

License GPL-3

URL https://github.com/mohsaqr/psychnet

BugReports https://github.com/mohsaqr/psychnet/issues

Additional_repositories https://mohsaqr.r-universe.dev

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports parallel, stats

Suggests cograph, glasso, glmnet, IsingFit, knitr, mgm, mvtnorm, psych, qgraph, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Repository https://mohsaqr.r-universe.dev

Date/Publication 2026-06-28 20:15:38 UTC

RemoteUrl https://github.com/mohsaqr/psychnet

RemoteRef HEAD

RemoteSha 9662904155844555943d8c79d91421ab777b380

Contents

`$.psychnet` *Back-compatible field access for a psychnet object*

Description

The canonical (`str`-visible) fields are the lean netobject set; this method adds virtual aliases so older/external accessors keep working without storing redundant fields.

Usage

```
## S3 method for class 'psychnet'
x$name
```

Arguments

`x`	A `psychnet` object.
`name`	Field name. Canonical fields plus the legacy aliases `graph` (= `weights`), `labels` (= `nodes$label`), `n_nodes`, `n_edges`, and `n_obs` (= `n`).

Value

The requested field, or `NULL` if neither a canonical field nor a known alias.

`as.data.frame.psychnet`
Tidy edge list for a psychnet network

Description

Tidy edge list for a psychnet network

Usage

```
## S3 method for class 'psychnet'
as.data.frame(x, row.names = NULL, optional = FALSE, ..., include_zero = FALSE)
```

Arguments

`x`	A `psychnet` object.
`row.names, optional`	Ignored (for S3 consistency).
`...`	Unused.
`include_zero`	If TRUE, keep zero-weight (absent) edges. Default FALSE.

Value

A one-row-per-edge `data.frame` with columns `from`, `to`, `weight`.

---

as.data.frame.psychnet_bootstrap *Tidy a network bootstrap*

Description

Tidy a network bootstrap

Usage

```
## S3 method for class 'psychnet_bootstrap'
as.data.frame(x, ...)
```

Arguments

`x`	A `psychnet_bootstrap` object.
`...`	Unused.

Value

The tidy `$edges` data frame (one row per edge, with its percentile interval, inclusion proportion, and `significant` flag).

---

certificate *Correctness certificate of a fitted network*

Description

Every regularized or constrained estimator in `psychnet` self-certifies: it reports how far the returned network sits from the unique optimum of its own convex objective (a KKT / stationarity residual), or – for the structural methods – whether the graph satisfies the identity that defines it. This verb returns that certificate as a tidy one-row `data.frame`, so correctness is read the same way for every method.

Usage

```
certificate(x, tol = 1e-06)
```

Arguments

`x`	A psychnet object.
`tol`	Tolerance below which the fit is flagged `certified = TRUE`. Default `1e-6`.

Details

The residual is near machine zero for a correctly solved problem. `cor` and `pcor` have no optimization to certify and report `NA`.

Value

A one-row `data.frame` with columns `method`, `certificate` (the residual; smaller is better), `kind` (`"kkt"` for the optimization certificates, `"structural"` for TMFG/relimp, `"none"` for cor/pcor), and `certified` (logical: residual at or below `tol`).

Examples

```
S <- 0.4^abs(outer(1:6, 1:6, "-"))
certificate(ebic_glasso(cor_matrix = S, n = 250))
certificate(tmfg_network(cor_matrix = S))
```

`condition` *Condition a moderated network at a moderator value*

Description

Extracts the effective pairwise network implied by a moderated MGM (`mgm_fit()` with `moderators`) at a given value of the moderator, mirroring `mgm::condition()`: it applies the AND-rule pre-filter, absorbs the moderator value into the main-effect coefficients, and re-aggregates the pairwise edges.

Usage

```
condition(object, value, rule = NULL)
```

Arguments

`object`	A `psychnet_moderated` object from `mgm_fit(..., moderators=)`.
`value`	Moderator value to condition on (e.g. `0` or `1` for a binary moderator, or any numeric for a continuous one).
`rule`	Symmetrization rule; defaults to the rule used at fit time.

Value

A `psychnet` network object (the moderator node carries no edges).

Examples

```
set.seed(1)
x1 <- stats::rnorm(400); x2 <- stats::rnorm(400)
mod <- rep(0:1, each = 200)
y <- x1 * (mod == 1) + stats::rnorm(400)   # x1-y edge only when mod == 1
d <- data.frame(x1 = x1, x2 = x2, y = y, mod = mod)
fit <- mgm_fit(d, types = c("g", "g", "g", "c"), moderators = 4)
condition(fit, value = 1)
```

cor_auto *Automatic correlation matrix (polychoric / polyserial / Pearson)*

Description

Detects ordinal variables (integer-valued with at most `ordinal_max_levels` levels) and returns the correlation matrix using a polychoric correlation for ordinal-ordinal pairs, a polyserial correlation for ordinal-continuous pairs, and Pearson otherwise, projected to the nearest positive-definite matrix. The base-R counterpart of `qgraph::cor_auto()`; this is the correlation `bootnet`/`qgraph` use by default for Likert data.

Usage

```
cor_auto(data, ordinal_max_levels = 7L, na_method = c("pairwise", "listwise"))
```

Arguments

`data`	Numeric data frame or matrix (rows = observations).
`ordinal_max_levels`	Maximum distinct values for a variable to count as ordinal. Default 7.
`na_method`	`"pairwise"` (default) or `"listwise"`.

Value

A correlation matrix with the variable names as dimnames.

Examples

```
set.seed(1)
z <- matrix(stats::rnorm(300 * 4), 300, 4) %*% chol(0.5^abs(outer(1:4, 1:4, "-")))
x <- apply(z, 2, function(col) as.integer(cut(col, 5)))   # 5-level Likert
cor_auto(x)
```

cor_network *Correlation network*

Description

Marginal (zero-order) association network: the Pearson correlation matrix with the diagonal removed. Equivalent to `bootnet`'s `"cor"` default.

Usage

```
cor_network(
  data = NULL,
  cor_matrix = NULL,
  n = NULL,
  cor_method = c("pearson", "spearman", "kendall", "auto"),
  threshold = 0,
  alpha = NULL,
  adjust = "none",
  na_method = c("pairwise", "listwise"),
  labels = NULL
)
```

Arguments

data	Numeric data frame or matrix (rows = observations). Optional if cor_matrix is supplied.
cor_matrix	Optional precomputed correlation matrix; if given, data is ignored and n is required when alpha is used.
n	Sample size (needed for significance testing when cor_matrix is supplied).
cor_method	Correlation method: "pearson" (default), "spearman", "kendall", or "auto" (polychoric/polyserial for ordinal items, the qgraph::cor_auto default; see cor_auto()).
threshold	Correlations with absolute value below this are set to zero. Default 0.
alpha	Significance level; if set, correlations not significant at alpha are zeroed. NULL (default) keeps every edge.
adjust	Multiple-comparison adjustment for the edge p-values (any stats::p.adjust method). Default "none".
na_method	Missing-data handling: "pairwise" (default) uses pairwise-complete correlations projected to the nearest positive-definite matrix; "listwise" drops rows with any NA. Identical when data is complete.
labels	Optional node labels.

Value

A psychnet object whose $weights is the thresholded correlation matrix, with $cor_matrix, $n_eff, $na_method (and $p_values when alpha is used).

Examples

```
x <- matrix(stats::rnorm(200 * 4), 200, 4)
cor_network(x)
cor_network(x, alpha = 0.05, adjust = "BH")
```

dichotomize *Dichotomize numeric columns to 0/1*

Description

Splits each column of a numeric matrix or data frame into a binary 0/1 variable. This is the usual preprocessing step before fitting an Ising network (`ising_fit()`, `ising_sampler()`) to Likert or other ordinal/continuous data, which require binary input.

Usage

```
dichotomize(data, method = c("median", "mean", "rank"))
```

Arguments

`data`	Numeric matrix or data frame (rows = observations).
`method`	Split rule, applied independently to each column:
	`"median"` (default) 1 if the value is >= the column median.
	`"mean"` 1 if the value is > the column mean.
	`"rank"` 1 for the upper half of the column by rank, giving a balanced (~50/50) split that is robust to ties (useful for coarse Likert items where a median split is badly unbalanced).

Value

An integer matrix of `0`/`1` values with the same dimensions and dimnames as `data`.

Examples

```
b <- dichotomize(SRL_GPT, method = "median")
table(b)                          # values are 0/1 only
```

difference_test *Bootstrapped difference test for edges or centralities*

Description

Tests, within a single network, whether two edge weights or two node centralities differ. For every pair it forms the per-resample difference from the stored bootstrap draws, takes the percentile interval of that difference, and flags the pair `significant` when the interval excludes zero; it also reports the two-sided bootstrap p-value (Epskamp, Borsboom & Fried 2018). This is the within-network counterpart to the edge accuracy intervals reported by `net_boot()`.

Usage

```
difference_test(boot, type = "edge", ci = NULL, p_adjust = "none")
```

Arguments

boot	A psychnet_bootstrap object from net_boot().
type	Quantity to compare: "edge" (default), or any centrality measure bootstrapped by net_boot() (e.g. "strength", "expected_influence").
ci	Confidence level for the difference interval. Defaults to the level used by the bootstrap object.
p_adjust	Multiple-comparison adjustment for the pairwise p-values (any stats::p.adjust method). Default "none".

Value

A tidy data frame, one row per pair, with item1, item2, the two observed values, their observed difference, the percentile interval of the bootstrap difference (lower, upper), the two-sided p_value, and a logical significant.

Examples

```
set.seed(1)
x <- matrix(stats::rnorm(150 * 5), 150, 5) %*% chol(0.4^abs(outer(1:5, 1:5, "-")))
colnames(x) <- paste0("V", 1:5)
bs <- net_boot(x, n_boot = 100)
difference_test(bs, type = "strength")
```

---

ebic_glasso *EBIC-regularized Gaussian graphical model (graphical lasso)*

Description

Selects an L1 penalty by the extended BIC (Foygel & Drton 2010) over a log-spaced path, then refits the chosen penalty to machine precision so the returned network is the certified global optimum of the convex objective. Equivalent in purpose to qgraph::EBICglasso() / bootnet's "EBICglasso" default, but pure base R and self-certified (see glasso_kkt()).

Usage

```
ebic_glasso(
  data = NULL,
  cor_matrix = NULL,
  n = NULL,
  gamma = 0.5,
  nlambda = 100L,
  lambda_min_ratio = 0.01,
  threshold = 0,
  cor_method = c("pearson", "spearman", "kendall", "auto"),
  na_method = c("pairwise", "listwise"),
  native = TRUE,
  labels = NULL
)
```

Arguments

data	Numeric data frame or matrix (rows = observations). Optional if cor_matrix is supplied.
cor_matrix	Optional correlation matrix; if given, n is required and data is ignored.
n	Sample size (required when cor_matrix is supplied).
gamma	EBIC hyperparameter. Default 0.5.
nlambda	Number of penalties on the path. Default 100.
lambda_min_ratio	Smallest penalty as a fraction of the largest. Default 0.01.
threshold	Partial correlations with absolute value below this are set to zero. Default 0.
cor_method	Correlation used when data is supplied: "pearson" (default), "spearman", "kendall", or "auto" (polychoric/polyserial for ordinal items, the qgraph::cor_auto / bootnet default). See cor_auto().
na_method	Missing-data handling when data is supplied: "pairwise" (default, pairwise-complete correlations + nearest-PD projection) or "listwise" (drop incomplete rows). Identical for complete data.
native	Solver switch. TRUE (default) uses psychnet's own pure-R, dependency-free, self-certified solver. FALSE delegates each fixed-penalty solve to the established glasso Fortran package (in Suggests) for speed and byte-identical glasso/qgraph output, at its looser convergence (the reported $kkt then shows glasso's tolerance rather than ~1e-11).
labels	Optional node labels.

Value

A psychnet object whose $weights is the partial-correlation matrix, with $precision, $lambda, $gamma, $cor_matrix, $ebic, $native, and $kkt (the stationarity residual of the returned network).

Examples

```
S <- 0.4^abs(outer(1:6, 1:6, "-"))
fit <- ebic_glasso(cor_matrix = S, n = 250)
fit
as.data.frame(fit)
```

---

ggm_modselect *Stepwise Gaussian graphical model selection (ggmModSelect)*

Description

Selects a GGM by extended-BIC model search over edge sets generated from the glasso path, refitting the *unregularized* maximum-likelihood precision on each candidate graph, with an optional stepwise add/drop search. Unlike the graphical lasso, retained edges are not shrunk. Equivalent in purpose to qgraph::ggmModSelect(), but pure base R and self-certified via ggm_support_kkt().

Usage

```
ggm_modselect(
  data = NULL,
  cor_matrix = NULL,
  n = NULL,
  gamma = 0,
  stepwise = TRUE,
  nlambda = 100L,
  lambda_min_ratio = 0.01,
  threshold = 0,
  cor_method = c("pearson", "spearman", "kendall", "auto"),
  na_method = c("pairwise", "listwise"),
  native = TRUE,
  labels = NULL
)
```

Arguments

`data`	Numeric data frame or matrix (rows = observations). Optional if `cor_matrix` is supplied.
`cor_matrix`	Optional correlation matrix; if given, `n` is required.
`n`	Sample size (required when `cor_matrix` is supplied).
`gamma`	EBIC hyperparameter. Default `0`, matching `qgraph::ggmModSelect()`: because the selected graph is refit with the *unregularized* MLE (no edge shrinkage), the extra EBIC penalty is not needed, so gamma 0 (plain BIC) is the method's intended setting. (The regularized GGMs `ebic_glasso()` and `huge_network()` keep gamma 0.5.)
`stepwise`	If `TRUE` (default), refine the best glasso-path graph by a greedy single-edge add/drop search.
`nlambda`	Number of glasso penalties scanned for candidate graphs.
`lambda_min_ratio`	Smallest penalty as a fraction of the largest.
`threshold`	Partial correlations with absolute value below this are zeroed. Default 0.
`cor_method`	Correlation used when `data` is supplied: `"pearson"` (default), `"spearman"`, `"kendall"`, or `"auto"` (polychoric/polyserial, the qgraph/bootnet default for ordinal items). See `cor_auto()`.
`na_method`	Missing-data handling when `data` is supplied: `"pairwise"` (default) or `"listwise"`. See `ebic_glasso()`.
`native`	Solver switch for generating candidate supports: `TRUE` (default) uses the pure-R solver; `FALSE` delegates to the `glasso` Fortran package (in `Suggests`). The reported precision is the unregularized refit either way. See `ebic_glasso()`.
`labels`	Optional node labels.

Value

A `psychnet` object whose `$weights` is the partial-correlation matrix, with `$precision`, `$support` (the selected graph), `$gamma`, `$ebic`, `$cor_matrix`, and `$kkt`.

Examples

```
S <- 0.5^abs(outer(1:6, 1:6, "-"))
ggm_modselect(cor_matrix = S, n = 250)
```

`ggm_support_kkt` *Constrained Gaussian-MRF (graph-restricted MLE) stationarity residual*

Description

Certificate for an *unregularized* Gaussian graphical model whose precision is constrained to a fixed graph (the estimator behind `ggm_modselect()` and `logo_network()`). The maximum-likelihood / maximum-entropy conditions for a Gaussian Markov random field on a graph G are exact: $W_{ij} = S_{ij}$ for every (i, j) on the graph and on the diagonal ($W = \Theta^{-1}$), and $\Theta_{ij} = 0$ for every (i, j) not on the graph. A near-zero return certifies the constrained optimum with no reference solver.

Usage

```
ggm_support_kkt(theta, cor_matrix, support, active_tol = 1e-08)
```

Arguments

`theta`	Precision matrix to test.
`cor_matrix`	Correlation / covariance the model was fit to.
`support`	Logical p x p matrix; `TRUE` where an edge is allowed.
`active_tol`	Magnitude above which an off-support entry counts as a nonzero violation.

Value

Maximum absolute stationarity violation (scalar); 0 = exact optimum.

Examples

```
S <- 0.4^abs(outer(1:6, 1:6, "-"))
fit <- ggm_modselect(cor_matrix = S, n = 250)
ggm_support_kkt(fit$precision, S, fit$support)
```

glasso_kkt *Graphical-lasso stationarity (KKT) residual*

Description

A dependency-free correctness certificate for a fitted Gaussian graphical model. For the convex objective

$$\min_{\Theta \succ 0} -\log\det\Theta + \operatorname{tr}(S\Theta) + \rho \sum_{i \neq j} |\Theta_{ij}|$$

(off-diagonal penalty), let $W = \Theta^{-1}$. The subgradient optimality conditions are $W_{ii} = S_{ii}$; $W_{ij} - S_{ij} = \rho\,\mathrm{sign}(\Theta_{ij})$ where $\Theta_{ij} \neq 0$; and $|W_{ij} - S_{ij}| \leq \rho$ otherwise. By strict convexity, a precision matrix with zero violation is the unique global optimum, so a near-zero return certifies correctness independently of any reference solver.

Usage

```
glasso_kkt(theta, cor_matrix, rho, active_tol = 1e-08)
```

Arguments

`theta`	Precision matrix to test.
`cor_matrix`	Correlation / covariance the model was fit to.
`rho`	Scalar penalty.
`active_tol`	Magnitude above which an off-diagonal entry is "active".

Value

Maximum absolute stationarity violation (scalar); 0 = exact optimum.

Examples

```
S <- 0.5^abs(outer(1:5, 1:5, "-"))
fit <- ebic_glasso(cor_matrix = S, n = 200)
glasso_kkt(fit$precision, S, fit$lambda)
```

glm_lasso_kkt *Stationarity (KKT) residual of an L1-penalized GLM fit*

Description

Dependency-free correctness certificate for a nodewise lasso, analogous to `glasso_kkt()` for the graphical lasso. With standardized predictors `X` and fitted mean `mu` (identity link for gaussian, logistic for binomial), the subgradient conditions are $n^{-1}X_j^\top(y-\mu) = \lambda\,\mathrm{sign}(\beta_j)$ for active coordinates and $|n^{-1}X_j^\top(y-\mu)| \leq \lambda$ otherwise. Near-zero certifies the penalized-likelihood optimum.

Usage

```
glm_lasso_kkt(
  X,
  y,
  b0,
  beta,
  lambda,
  family = "gaussian",
  weights = NULL,
  active_tol = 1e-08
)
```

Arguments

X	Standardized predictor matrix (mean 0, unit variance columns).
y	Response.
b0	Fitted intercept.
beta	Fitted (standardized) coefficients.
lambda	Penalty.
family	"gaussian" or "binomial".
weights	Optional observation weights (NULL = unweighted).
active_tol	Magnitude above which a coefficient is "active".

Value

Maximum absolute stationarity violation (scalar). Near-zero certifies the fit is at the penalized-likelihood optimum.

Examples

```
set.seed(1)
x <- scale(matrix(stats::rnorm(200 * 3), 200, 3))
y <- as.numeric(x %*% c(0.5, 0, -0.3) + stats::rnorm(200))
fit <- stats::lm.fit(cbind(1, x), y)
glm_lasso_kkt(x, y, fit$coefficients[1], fit$coefficients[-1], lambda = 0)
```

---

huge_network *Nonparanormal graphical model (huge)*

Description

Estimates a Gaussian graphical model after a rank-based nonparanormal transform that relaxes the multivariate-normal assumption, then selects the L1 penalty by EBIC and refits to the certified optimum. Equivalent in purpose to huge::huge() (nonparanormal) / bootnet's "huge" default, but pure base R and self-certified via glasso_kkt() on the transformed correlation.

Usage

```
huge_network(
  data = NULL,
  cor_matrix = NULL,
  n = NULL,
  npn = c("shrinkage", "truncation", "skeptic"),
  gamma = 0.5,
  nlambda = 100L,
  lambda_min_ratio = 0.01,
  threshold = 0,
  na_method = c("pairwise", "listwise"),
  native = TRUE,
  labels = NULL
)
```

Arguments

`data`	Numeric data frame or matrix (rows = observations). Optional if `cor_matrix` is supplied (then the transform is skipped).
`cor_matrix`	Optional pre-transformed correlation matrix; if given, `n` is required, `data` and `npn` are ignored.
`n`	Sample size (required when `cor_matrix` is supplied).
`npn`	Nonparanormal transform: `"shrinkage"` (default), `"truncation"`, or `"skeptic"` (Spearman).
`gamma`	EBIC hyperparameter. Default 0.5.
`nlambda`	Number of penalties on the path. Default 100.
`lambda_min_ratio`	Smallest penalty as a fraction of the largest.
`threshold`	Partial correlations with absolute value below this are zeroed. Default 0.
`na_method`	Missing-data handling when `data` is supplied: `"pairwise"` (default, with the nonparanormal transform applied per column over observed values) or `"listwise"`. See `ebic_glasso()`.
`native`	Solver switch for the glasso path: `TRUE` (default) uses the pure-R solver; `FALSE` delegates to the `glasso` Fortran package (in `Suggests`). See `ebic_glasso()`.
`labels`	Optional node labels.

Value

A `psychnet` object whose `$weights` is the partial-correlation matrix, with `$precision`, `$lambda`, `$gamma`, `$cor_matrix` (the transformed correlation), `$npn`, `$ebic`, and `$kkt`.

Examples

```
set.seed(1)
x <- matrix(stats::rnorm(300 * 5), 300, 5)
x <- exp(x %*% chol(0.4^abs(outer(1:5, 1:5, "-"))))   # break normality
huge_network(x)
```

`ising_fit` *Ising network for binary data*

Description

Estimates an Ising model by nodewise L1-penalized logistic regression with EBIC selection, combined by the AND (default) or OR rule. Equivalent in purpose to `IsingFit::IsingFit()`, but pure base R and self-certified: each node's regression reports its stationarity (KKT) residual (see `glm_lasso_kkt()`).

Usage

```
ising_fit(
  data,
  gamma = 0.25,
  rule = c("AND", "OR"),
  nlambda = 100L,
  lambda_min_ratio = 0.01,
  min_sum = NULL,
  weights = NULL,
  na_method = c("pairwise", "listwise"),
  native = TRUE,
  labels = NULL
)
```

Arguments

`data`	Binary (0/1) data frame or matrix (rows = observations).
`gamma`	EBIC hyperparameter. Default 0.25.
`rule`	Edge-combination rule: `"AND"` (default) or `"OR"`.
`nlambda`	Number of penalties per nodewise path. Default 100.
`lambda_min_ratio`	Smallest penalty as a fraction of the largest.
`min_sum`	Minimum row sum-score (number of endorsed items); rows below it are dropped before fitting. `NULL` (default) keeps every row.
`weights`	Optional non-negative observation weights, one per retained row. `NULL` (default) is unweighted.
`na_method`	Missing-data handling: `"pairwise"` (default) single-imputes each column over its observed values (mode for binary), keeping the full sample; `"listwise"` drops incomplete rows. Identical for complete data.
`native`	Solver switch. `TRUE` (default) uses psychnet's own pure-R, dependency-free, self-certified L1 logistic path (KKT ~1e-9). `FALSE` delegates each per-node fit to the `glmnet` package with the IsingFit EBIC path, so the returned `$weights`/`$thresholds` byte-match `IsingFit::IsingFit()` (to ~1e-16) at the cost of glmnet's looser self-certificate. `native = FALSE` needs the optional `glmnet` package (Suggests); `weights`/`min_sum` are supported with `native = TRUE` only.

labels	Optional node labels.

Value

A psychnet object whose $weights is the symmetric weight matrix, with $thresholds (node intercepts) and $kkt (the worst nodewise stationarity residual).

Examples

```
set.seed(1)
z <- matrix(stats::rnorm(400 * 2), 400, 2)
x <- cbind(z[, 1], z[, 1], z[, 2], z[, 2]) + matrix(stats::rnorm(400 * 4), 400)
b <- (x > 0) * 1L
colnames(b) <- paste0("V", 1:4)
ising_fit(b)
```

---

ising_sampler *Unregularized Ising network for binary data*

Description

Estimates an Ising model by *unpenalized* nodewise logistic regression, with optional Wald p-value edge pruning, combined by the AND (default) or OR rule. The unregularized counterpart of ising_fit(); self-certified by the maximum-likelihood score residual (see glm_lasso_kkt() at lambda = 0).

Usage

```
ising_sampler(
  data,
  rule = c("AND", "OR"),
  alpha = NULL,
  adjust = "none",
  min_sum = NULL,
  weights = NULL,
  na_method = c("pairwise", "listwise"),
  labels = NULL
)
```

Arguments

data	Binary (0/1) data frame or matrix (rows = observations).
rule	Edge-combination rule: "AND" (default) or "OR".
alpha	Significance level for Wald edge pruning; NULL (default) keeps every edge.
adjust	Multiple-comparison adjustment for the edge p-values (any stats::p.adjust method). Default "none".

min_sum	Minimum row sum-score; rows below it are dropped before fitting. NULL (default) keeps every row.
weights	Optional non-negative observation weights, one per retained row. NULL (default) is unweighted.
na_method	Missing-data handling: "pairwise" (default, mode-impute) or "listwise". See ising_fit().
labels	Optional node labels.

Value

A psychnet object whose $weights is the symmetric weight matrix, with $thresholds (node intercepts), $rule, $p_values, $nodewise (for net_predict()), and $kkt (worst nodewise score residual).

Examples

```
set.seed(1)
z <- matrix(stats::rnorm(500 * 2), 500, 2)
x <- cbind(z[, 1], z[, 1], z[, 2], z[, 2]) + matrix(stats::rnorm(500 * 4), 500)
b <- (x > 0) * 1L
colnames(b) <- paste0("V", 1:4)
ising_sampler(b)
```

---

lmg_certificate *Relative-importance (LMG / Shapley) certificate*

Description

By Shapley efficiency, the importance shares a node receives from its predictors sum exactly to that node's full-model R-squared. Returns the maximum absolute deviation from that identity; near zero certifies the decomposition.

Usage

```
lmg_certificate(x)
```

Arguments

x	A psychnet object produced by relimp_network().

Value

Maximum absolute deviation of incoming-share sums from the per-node R-squared (scalar); 0 = exact decomposition.

Examples

```
S <- 0.4^abs(outer(1:5, 1:5, "-"))
lmg_certificate(relimp_network(cor_matrix = S))
```

logo_network *Local-Global sparse inverse covariance (LoGo)*

Description

Estimates a sparse Gaussian graphical model whose conditional-independence structure is the chordal TMFG: the precision is the closed-form Gaussian Markov random field on that graph (Barfuss et al. 2016). Equivalent in purpose to `NetworkToolbox::LoGo()` / `bootnet`'s `"LoGo"` default, pure base R and self-certified via `ggm_support_kkt()` (the precision reproduces `S` exactly on the TMFG support).

Usage

```
logo_network(
  data = NULL,
  cor_matrix = NULL,
  n = NULL,
  cor_method = c("pearson", "spearman", "kendall", "auto"),
  threshold = 0,
  na_method = c("pairwise", "listwise"),
  labels = NULL
)
```

Arguments

`data`	Numeric data frame or matrix (rows = observations). Optional if `cor_matrix` is supplied.
`cor_matrix`	Optional correlation matrix; if given, `n` is required.
`n`	Sample size (recorded on the result; required with `cor_matrix`).
`cor_method`	Correlation when `data` is supplied: `"pearson"` (default), `"spearman"`, `"kendall"`, or `"auto"` (polychoric/polyserial; see `cor_auto()`).
`threshold`	Partial correlations with absolute value below this are zeroed. Default 0.
`na_method`	Missing-data handling when `data` is supplied: `"pairwise"` (default) or `"listwise"`. See `ebic_glasso()`.
`labels`	Optional node labels.

Value

A `psychnet` object whose `$weights` is the partial-correlation matrix, with `$precision`, `$support` (the TMFG graph), `$cor_matrix`, and `$kkt`.

Examples

```
set.seed(1)
x <- matrix(stats::rnorm(300 * 6), 300, 6)
logo_network(x)
```

`mgm_fit` *Mixed graphical model*

Description

Estimates a mixed graphical model by nodewise L1-penalized regression – a gaussian (linear) lasso for continuous nodes and a logistic lasso for binary nodes – with per-node EBIC selection, combined by the AND rule. Equivalent in purpose to `mgm::mgm()`, but pure base R and self-certified: each node's regression reports its stationarity (KKT) residual (see `glm_lasso_kkt()`).

Usage

```
mgm_fit(
  data,
  gamma = 0.25,
  types = NULL,
  nlambda = 100L,
  lambda_min_ratio = 0.01,
  threshold = c("LW", "HW", "none"),
  rule = c("AND", "OR"),
  moderators = NULL,
  weights = NULL,
  na_method = c("pairwise", "listwise"),
  native = TRUE,
  labels = NULL
)
```

Arguments

`data`	Numeric data frame or matrix (rows = observations); columns are continuous or binary (0/1).
`gamma`	EBIC hyperparameter. Default 0.25.
`types`	Optional character vector of node types (`"g"` gaussian, `"c"` binary); auto-detected if `NULL`.
`nlambda`	Number of penalties per nodewise path. Default 100.
`lambda_min_ratio`	Smallest penalty as a fraction of the largest.
`threshold`	Post-selection coefficient threshold: `"LW"` (default), `"HW"`, or `"none"`, matching `mgm::mgm()`.
`rule`	Edge-combination rule: `"AND"` (default) or `"OR"`.
`moderators`	Optional single column index of a moderator variable. When supplied, fits a *moderated* MGM (that variable moderates every pairwise edge) and returns a `psychnet_moderated` object to be read with `condition()`; glmnet-based, and `weights` are not supported in this mode.

weights	Optional non-negative observation weights, one per row of the (NA-prepared) data. NULL (default) is unweighted.
na_method	Missing-data handling: "pairwise" (default) single-imputes each column over its observed values (mean for continuous, mode for binary), keeping the full sample; "listwise" drops incomplete rows.
native	Solver switch. TRUE (default) uses psychnet's own pure-R, dependency-free, self-certified L1 path (KKT ~1e-9). FALSE delegates each per-node fit to the glmnet package with mgm's exact EBIC/LW path (gaussian lasso for continuous nodes, 2-class multinomial lasso for binary nodes), so the returned edge magnitudes byte-match abs(mgm::mgm()$pairwise$wadj) (to ~1e-6) at the cost of glmnet's looser self-certificate. native = FALSE needs the optional glmnet package (Suggests); weights are supported with native = TRUE only.
labels	Optional node labels.

Value

A psychnet object whose $weights is the symmetric standardized weight matrix, with $types and $kkt (the worst nodewise residual). A binary-binary edge carries the sign of its nodewise-logistic coefficient; mgm::mgm() reports the same edge as a magnitude only (its sign is undefined for a categorical-categorical interaction), so compare such edges on abs(). Continuous columns are standardized internally, binary predictors enter the graph on their 0/1 dummy scale, and binary-response logit coefficients are converted to mgm's two-class multinomial scale before edge aggregation. With these conventions the edge magnitudes match mgm::mgm closely for gaussian-gaussian, gaussian-binary, and binary-binary edges alike; weak edges near the EBIC/threshold boundary can still differ in support because the penalty is selected on an independent base-R path.

Examples

```
set.seed(1)
f <- stats::rnorm(400)
g1 <- f + stats::rnorm(400); g2 <- f + stats::rnorm(400)
b1 <- (f + stats::rnorm(400) > 0) * 1L
d <- data.frame(g1 = g1, g2 = g2, b1 = b1, n = stats::rnorm(400))
mgm_fit(d)
```

---

net_boot *Bootstrap a psychometric network*

Description

Resamples observations with replacement, re-estimates the network on each resample, and summarizes the sampling distribution of every edge weight and node centrality (mean, percentile confidence interval, and edge inclusion proportion). An edge is flagged significant when its percentile interval excludes zero. The raw per-resample draws are stored on the returned object for use by difference_test().

Usage

```
net_boot(
  data,
  method = "glasso",
  n_boot = 1000L,
  ci = 0.95,
  measures = c("strength", "expected_influence"),
  centrality_fn = NULL,
  predictability = FALSE,
  threshold = FALSE,
  diff_test = FALSE,
  p_adjust = "none",
  labels = NULL,
  cores = NULL,
  engine = NULL,
  ...
)
```

Arguments

`data`	Numeric data frame or matrix (rows = observations).
`method`	Estimator (see `psychnet()`). Default `"glasso"`.
`n_boot`	Number of bootstrap resamples. Default 1000.
`ci`	Confidence level for percentile intervals. Default 0.95.
`measures`	Centrality measures to bootstrap. Defaults to the two recommended for psychometric networks (`"strength"`, `"expected_influence"`); `"betweenness"`/`"closeness"` and custom measures (via `centrality_fn`) are also accepted. See `net_centralities()`.
`centrality_fn`	Optional function supplying any non-built-in `measures` (see `net_centralities()`).
`predictability`	Logical; if `TRUE` and the estimator returns a precision matrix (GGM family), bootstrap node predictability (R^2) and report its interval. Default `FALSE`.
`threshold`	Logical; if `TRUE`, also return the observed network with every edge whose bootstrap interval includes zero set to zero (`$thresholded`). Default `FALSE`.
`diff_test`	Logical; if `TRUE`, also return two-sided bootstrap difference p-value matrices for edges (`$edge_diff_p`, `NULL` past 500 edges) and for each centrality measure (`$centrality_diff_p`). Default `FALSE`.
`p_adjust`	Multiple-comparison adjustment applied to the difference p-value matrices (any stats::p.adjust method). Default `"none"`.
`labels`	Optional node labels.
`cores`	Number of CPU cores for the resample loop. `NULL` (default) uses two thirds of the detected cores; 1 forces a serial run. Parallelism uses forking (`parallel::mclapply`) and falls back to serial on Windows. Because every resample index is drawn in the parent process before any fitting, the result is identical for any number of cores and reproducible from `set.seed()`.

engine	Optional estimator engine forwarded to each resample fit (e.g. "base"/"glasso" for glasso, "base"/"glmnet" for ising/mgm). NULL (default) uses the estimator's own default.
...	Passed to the estimator.

Value

An object of class psychnet_bootstrap: tidy $edges (with a significant flag) and $centrality data frames, the observed network in $observed, raw resample draws in $edge_boot, $str_boot, $ei_boot, and the general $centrality_boot (named list, one matrix per measure). Optional $predictability, $thresholded, $edge_diff_p, $centrality_diff_p, plus $lambda_path/$lambda_selected when the estimator reports them.

Examples

```
set.seed(1)
x <- matrix(stats::rnorm(150 * 5), 150, 5) %*% chol(0.4^abs(outer(1:5, 1:5, "-")))
colnames(x) <- paste0("V", 1:5)
bs <- net_boot(x, n_boot = 50, cores = 1)
as.data.frame(bs)
```

---

net_centralities *Node centrality*

Description

Node centrality

Usage

```
net_centralities(
  x,
  measures = c("strength", "expected_influence"),
  centrality_fn = NULL,
  ...
)
```

Arguments

x	A psychnet object or a weighted adjacency matrix.
measures	Character vector of measures to return. Any of "strength", "expected_influence" (the defaults, recommended for psychometric networks), "betweenness", "closeness", plus any names supplied via centrality_fn. Betweenness and closeness are computed on the absolute, inverted-weight graph and are not generally meaningful on signed networks – request them only when a downstream comparison needs them.
centrality_fn	Optional function taking the weighted adjacency matrix and returning a named list of node-centrality vectors, used to supply any measures not built in.
...	Unused.

Value

A tidy `data.frame`, one row per node, with a `node` column and one column per requested measure (`strength` = sum of absolute edge weights, `expected_influence` = sum of signed edge weights, by default).

Examples

```
S <- 0.4^abs(outer(1:6, 1:6, "-"))
net_centralities(ebic_glasso(cor_matrix = S, n = 250))
```

`net_compare` *Network Comparison Test*

Description

Permutation test for whether two groups' Gaussian graphical models differ, on three invariants: global strength (`M`), maximum edge difference (`S`), and per-edge differences (`E`). Networks are EBIC graphical lassos (clean-room pure R). Equivalent in purpose to `NetworkComparisonTest::NCT()`.

Usage

```
net_compare(
  data1,
  data2,
  iter = 1000L,
  gamma = 0.5,
  paired = FALSE,
  abs = TRUE,
  weighted = TRUE,
  p_adjust = "none"
)
```

Arguments

`data1`, `data2`	Numeric data frames/matrices with the same columns.
`iter`	Number of permutations. Default 1000.
`gamma`	EBIC hyperparameter. Default 0.5.
`paired`	Logical; within-row swapping for paired designs. Default FALSE.
`abs`	Logical; compare absolute edge weights. Default TRUE.
`weighted`	Logical; if FALSE, binarize networks first. Default TRUE.
`p_adjust`	Multiple-comparison adjustment for per-edge p-values (any stats::p.adjust method). Default `"none"`.

Value

An object of class `psychnet_nct` with `$nw1`, `$nw2`, and `$M`, `$S`, `$E` (each `observed`, `perm`, `p_value`); `$E` also carries `edge_names`, a `from`/`to` data frame aligned to the per-edge vector.

Examples

```
set.seed(1)
a <- matrix(stats::rnorm(150 * 5), 150, 5)
b <- matrix(stats::rnorm(150 * 5), 150, 5)
colnames(a) <- colnames(b) <- paste0("V", 1:5)
fit <- net_compare(a, b, iter = 50)
fit
```

---

net_crosswalk *Argument crosswalk: psychnet as a substitute for qgraph / IsingFit / mgm*

Description

A tidy, one-row-per-argument map from each reference package's estimator to its psychnet equivalent, so users migrating from qgraph::EBICglasso, qgraph::cor_auto, qgraph::ggmModSelect, IsingFit::IsingFit, or mgm::mgm can find the matching argument and see what psychnet changes by default. Cross-sectional estimators only (temporal models are out of scope).

Usage

```
net_crosswalk(
  reference = c("all", "EBICglasso", "cor_auto", "ggmModSelect", "IsingFit", "mgm")
)
```

Arguments

reference	Which reference function to show: "all" (default), "EBICglasso", "cor_auto", "ggmModSelect", "IsingFit", or "mgm".

Value

A tidy data.frame, one row per argument, with columns reference (the pkg::fn being substituted), psychnet (the psychnet verb), ref_arg, psychnet_arg ("-" when there is no counterpart), status (identical / renamed / default differs / semantics differ / reference only / psychnet only), and a short note.

Examples

```
net_crosswalk("EBICglasso")
net_crosswalk("IsingFit")
```

net_predict *Node predictability*

Description

Reports how well each node is predicted by the others in a fitted network. For Gaussian graphical models this is the closed-form variance explained (R-squared) from the precision matrix and needs no data. For the nodewise models (`ising_fit()`, `ising_sampler()`, `mgm_fit()`) it requires the data and reports R-squared for Gaussian nodes and classification accuracy (`CC`) plus normalized accuracy (`nCC`) for binary nodes.

Usage

```
net_predict(x, data = NULL, ...)
```

Arguments

`x`	A psychnet object.
`data`	The data the network was estimated from; required for the nodewise models (ising / IsingSampler / mgm), ignored for the GGMs.
`...`	Unused.

Value

A tidy `data.frame`, one row per node, with columns `node`, `type` (`"gaussian"` or `"binary"`), `metric` (`"R2"` or `"nCC"`), `predictability`, and `accuracy` (classification accuracy for binary nodes, `NA` for Gaussian).

Examples

```
S <- 0.4^abs(outer(1:6, 1:6, "-"))
net_predict(ebic_glasso(cor_matrix = S, n = 250))
```

net_stability *Centrality-stability coefficient (case-dropping subset bootstrap)*

Description

Centrality-stability coefficient (case-dropping subset bootstrap)

Usage

```
net_stability(
  data,
  method = "glasso",
  measures = c("strength", "expected_influence"),
  centrality_fn = NULL,
  drop_prop = seq(0.1, 0.9, by = 0.1),
  iter = 100L,
  threshold = 0.7,
  certainty = 0.95,
  labels = NULL,
  ...
)
```

Arguments

`data`	Numeric data frame or matrix (rows = observations).
`method`	Estimator (see `psychnet()`). Default `"glasso"`.
`measures`	Centrality measures to assess. Defaults to the two recommended for psychometric networks (`c("strength", "expected_influence")`); `"betweenness"`/`"closeness"` and custom measures (via `centrality_fn`) are also accepted. See `net_centralities()`.
`centrality_fn`	Optional function supplying any non-built-in `measures` (see `net_centralities()`).
`drop_prop`	Proportions of cases to drop. Default `seq(0.1, 0.9, 0.1)`.
`iter`	Subsets per proportion. Default 100.
`threshold`	Minimum acceptable rank correlation. Default 0.7.
`certainty`	Probability the correlation must exceed `threshold`. Default 0.95.
`labels`	Optional node labels.
`...`	Passed to the estimator.

Value

An object of class `psychnet_stability` with `$cs` (CS-coefficient per measure) and a tidy `$table` of mean correlations by drop proportion.

Examples

```
set.seed(1)
x <- matrix(stats::rnorm(200 * 5), 200, 5) %*% chol(0.4^abs(outer(1:5, 1:5, "-")))
colnames(x) <- paste0("V", 1:5)
cs <- net_stability(x, drop_prop = c(0.3, 0.5, 0.7), iter = 20)
cs$cs
```

node_predictability *Node predictability as a plotting vector*

Description

A thin companion to net_predict() that returns predictability as a plain numeric vector in node order, clamped to [0, 1] – the form cograph::splot() expects for pie_values (the predictability ring drawn around each node). Use net_predict() when you want the full tidy table.

Usage

```
node_predictability(x, data = NULL)
```

Arguments

x	A psychnet object.
data	The data the network was estimated from; required for the nodewise models (ising / ising_sampler / mgm), ignored for the GGMs.

Value

A named numeric vector, one value per node (node order), each in [0, 1].

Examples

```
S <- 0.4^abs(outer(1:6, 1:6, "-"))
node_predictability(ebic_glasso(cor_matrix = S, n = 250))
```

pcor_network *Partial correlation network*

Description

Conditional (full-order) association network: each edge is the correlation between two variables with all others partialled out, obtained from the inverse correlation matrix. Equivalent to bootnet's "pcor" default.

Usage

```
pcor_network(
  data = NULL,
  cor_matrix = NULL,
  n = NULL,
  cor_method = c("pearson", "spearman", "kendall", "auto"),
  threshold = 0,
  alpha = NULL,
```

```
  adjust = "none",
  na_method = c("pairwise", "listwise"),
  labels = NULL
)
```

Arguments

data	Numeric data frame or matrix (rows = observations). Optional if cor_matrix is supplied.
cor_matrix	Optional precomputed correlation matrix; if given, data is ignored and n is required when alpha is used.
n	Sample size (needed for significance testing when cor_matrix is supplied).
cor_method	Correlation method: "pearson" (default), "spearman", "kendall", or "auto" (polychoric/polyserial for ordinal items, the qgraph::cor_auto default; see cor_auto()).
threshold	Correlations with absolute value below this are set to zero. Default 0.
alpha	Significance level; if set, correlations not significant at alpha are zeroed. NULL (default) keeps every edge.
adjust	Multiple-comparison adjustment for the edge p-values (any stats::p.adjust method). Default "none".
na_method	Missing-data handling: "pairwise" (default) uses pairwise-complete correlations projected to the nearest positive-definite matrix; "listwise" drops rows with any NA. Identical when data is complete.
labels	Optional node labels.

Value

A psychnet object whose $weights is the thresholded partial-correlation matrix, with $precision, $cor_matrix (and $p_values when alpha is used).

Examples

```
x <- matrix(stats::rnorm(200 * 4), 200, 4)
pcor_network(x)
pcor_network(x, alpha = 0.05, adjust = "holm")
```

---

print.psychnet *Print a psychnet network*

Description

Print a psychnet network

Usage

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

Arguments

x	A psychnet object.
...	Unused.

Value

x, invisibly.

---

print.psychnet_bootstrap *Print a network bootstrap*

Description

Print a network bootstrap

Usage

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

Arguments

x	A psychnet_bootstrap object.
...	Unused.

Value

x, invisibly.

---

print.psychnet_moderated *Print a moderated MGM fit*

Description

Print a moderated MGM fit

Usage

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

Arguments

x	A psychnet_moderated object.
...	Unused.

Value

x, invisibly.

---

print.psychnet_nct *Print a Network Comparison Test*

Description

Print a Network Comparison Test

Usage

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

Arguments

x	A psychnet_nct object.
...	Unused.

Value

x, invisibly.

---

print.psychnet_stability *Print a centrality-stability result*

Description

Print a centrality-stability result

Usage

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

Arguments

x	A psychnet_stability object.
...	Unused.

Value

x, invisibly.

---

psychnet *Estimate a psychometric network*

Description

The package's main entry point: routes to the requested estimator and returns a common psychnet object, so callers can swap estimators without rewiring downstream code.

Usage

```
psychnet(
  data,
 method = c("glasso", "cor", "pcor", "ising", "mgm", "huge", "ggm", "tmfg", "logo",
    "relimp", "ising_sampler"),
  threshold = 0,
  gamma = NULL,
  labels = NULL,
  ...
)
```

Arguments

data	Numeric data frame or matrix (rows = observations).
method	Estimator. One of "glasso" (default), "ggm", "tmfg", "logo", "relimp", "ising", "ising_sampler", "huge", "mgm", "cor", "pcor". The qgraph/bootnet names are accepted as aliases (e.g. "EBICglasso" -> "glasso", "ggmModSelect" -> "ggm", "LoGo" -> "logo").
threshold	Absolute-weight threshold below which edges are zeroed (forwarded only to the methods that take it: cor, pcor, glasso, huge, ggm, logo).
gamma	EBIC hyperparameter. NULL (default) keeps each method's own default (0.5 for the regularized Gaussian graphical models, 0 for ggm, 0.25 for ising/mgm); set it to override. Forwarded only to the regularized methods.
labels	Optional node labels.
...	Passed to the underlying estimator (e.g. cor_method= for the correlation-based methods, npn= for "huge", rule= for the Ising methods, alpha= for the correlation / "ising_sampler" methods).

Details

method speaks the package's own short vocabulary – "glasso", "ggm", "tmfg", "logo", "relimp", "ising", "ising_sampler", "huge", "mgm", "cor", "pcor". For interoperability it **also** accepts the qgraph/bootnet spellings ("EBICglasso", "ggmModSelect", "TMFG", "LoGo", "IsingFit", "IsingSampler"), which resolve to the same estimators. Whichever you pass in, the stored $method is the short name.

Value

A psychnet object.

Examples

```
x <- matrix(stats::rnorm(200 * 5), 200, 5)
psychnet(x, method = "glasso")
psychnet(x, method = "pcor", cor_method = "spearman")
psychnet(x, method = "EBICglasso")  # qgraph alias, same result
```

---

relimp_network *Relative-importance network (LMG / Shapley)*

Description

Builds a directed network in which the edge predictor -> outcome is the predictor's LMG (Shapley) share of the outcome node's regression R-squared. Equivalent in purpose to `relaimpo::calc.relimp(type = "lmg")` applied nodewise / `bootnet`'s `"relimp"` default, pure base R and self-certified via `lmg_certificate()`.

Usage

```
relimp_network(
  data = NULL,
  cor_matrix = NULL,
  cor_method = c("pearson", "spearman", "kendall", "auto"),
  max_nodes = 21L,
  na_method = c("pairwise", "listwise"),
  labels = NULL
)
```

Arguments

`data`	Numeric data frame or matrix (rows = observations). Optional if `cor_matrix` is supplied.
`cor_matrix`	Optional correlation matrix.
`cor_method`	Correlation when `data` is supplied: `"pearson"` (default), `"spearman"`, `"kendall"`, or `"auto"` (polychoric/polyserial; see `cor_auto()`).
`max_nodes`	Refuse to run above this many nodes (the cost grows as `2^(p-1)` per node). Default 21.
`na_method`	Missing-data handling when `data` is supplied: `"pairwise"` (default) or `"listwise"`. See `ebic_glasso()`.
`labels`	Optional node labels.

Value

A psychnet object whose $weights is the directed importance matrix (weights[k, j] = importance of k for outcome j), with $r2 (per-node full-model R-squared), $cor_matrix, and $kkt (the decomposition residual).

Examples

```
S <- 0.4^abs(outer(1:5, 1:5, "-"))
relimp_network(cor_matrix = S)
```

SRL *Self-regulated-learning (MSLQ) construct scores simulated by large language models*

Description

Five data sets of Motivated Strategies for Learning Questionnaire (MSLQ) construct scores, each a random sample of 300 cases generated by one large language model in the study *"Delving into the psychology of Machines: Exploring the structure of self-regulated learning via LLM-generated survey responses"* (Computers in Human Behavior, 2025). One data set per model: SRL_GPT, SRL_Gemini, SRL_Claude, SRL_Mistral, and SRL_LLaMa.

Usage

```
SRL_GPT

SRL_Gemini

SRL_Claude

SRL_Mistral

SRL_LLaMa
```

Format

Each is a data frame with 300 rows and 5 variables – the MSLQ construct scores (each the mean of that construct's Likert items, range 1–7):

CSU cognitive strategy use

IV intrinsic value

SE self-efficacy

SR self-regulation

TA test anxiety

Details

Only the five models whose responses carry estimable structure are included. Two other generators from the original study (ChatGPT and LeChat) produced near-independent items (mean absolute inter-item correlation ≈ 0.03), so their network is the empty graph; they are omitted. The five retained models span the spectrum the paper describes, from realistically structured (`SRL_GPT`) to strongly "over-coherent" (`SRL_Gemini`, `SRL_Claude`).

Source

Saqr, M. (2025). Delving into the psychology of Machines: Exploring the structure of self-regulated learning via LLM-generated survey responses. *Computers in Human Behavior*, 173, 108769. doi:10.1016/j.chb.2025.108769

Examples

```
# partial-correlation network of the five constructs for one model
net <- ebic_glasso(SRL_GPT)
net
net_centralities(net)
```

`summary.psychnet` *Summarize a psychnet network*

Description

Summarize a psychnet network

Usage

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

Arguments

`object`	A `psychnet` object.
`...`	Unused.

Value

The tidy edge list (invisibly); prints a summary as a side effect.

tmfg_certificate *Structural certificate for a TMFG network*

Description

A TMFG has no convex objective, so its correctness is certified structurally: a valid TMFG on p >= 3 nodes has exactly 3(p - 2) edges, is connected, and is chordal (every cycle of length >= 4 has a chord). Returns a non-negative score that is 0 for a valid TMFG.

Usage

```
tmfg_certificate(x)
```

Arguments

x	A psychnet object produced by tmfg_network().

Value

Scalar; 0 certifies a valid TMFG (correct edge count, connected, chordal), otherwise a positive integer counting the violated invariants.

Examples

```
set.seed(1)
x <- matrix(stats::rnorm(200 * 6), 200, 6)
tmfg_certificate(tmfg_network(x))
```

tmfg_network *Triangulated Maximally Filtered Graph (TMFG)*

Description

Builds a sparse, planar, chordal association network by greedily retaining the 3(p - 2) most informative edges (Massara et al. 2016). Equivalent in purpose to NetworkToolbox::TMFG() / bootnet's "TMFG" default, pure base R; correctness is certified structurally by tmfg_certificate().

Usage

```
tmfg_network(
  data = NULL,
  cor_matrix = NULL,
  cor_method = c("pearson", "spearman", "kendall", "auto"),
  na_method = c("pairwise", "listwise"),
  labels = NULL
)
```

Arguments

data	Numeric data frame or matrix (rows = observations). Optional if cor_matrix is supplied.
cor_matrix	Optional correlation matrix.
cor_method	Correlation when data is supplied: "pearson" (default), "spearman", "kendall", or "auto" (polychoric/polyserial; see cor_auto()).
na_method	Missing-data handling when data is supplied: "pairwise" (default) or "listwise". See ebic_glasso().
labels	Optional node labels.

Value

A psychnet object whose $weights is the filtered (signed) correlation matrix on the retained edges, with $adjacency, $cliques, $separators (the chordal decomposition used by logo_network()), and $cor_matrix.

Examples

```
set.seed(1)
x <- matrix(stats::rnorm(200 * 6), 200, 6)
tmfg_network(x)
```

Index