

Package: nda (via r-universe)

October 7, 2026

Type Package

Title Generalized Network-Based Dimensionality Reduction and Analysis

Version 0.3.0

Maintainer Zsolt T. Kosztian <kosztian.zsolt@gtk.uni-pannon.hu>

Description Constructs interpretable latent variables from communities in squared-association networks. Supports full, partial, and semipartial associations, feature selection, sparse networks, inference, prediction, visualization, and supervised regression. The generalized method is described in Kosztian et al. (2024) <[doi:10.1016/j.eswa.2023.121779](https://doi.org/10.1016/j.eswa.2023.121779)>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

LazyDataCompression xz

URL <https://github.com/kzst/nda>

BugReports <https://github.com/kzst/nda/issues>

Depends R (>= 4.0.0)

Imports energy, psych, stats, graphics, grDevices, igraph, Matrix, MASS, mco, parallel, ppcor, visNetwork

Suggests glmnet, plotly, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

Config/pak/sysreqs cmake libglpk-dev make libgsl0-dev libuv1-dev libxml2-dev

Repository <https://kzst.r-universe.dev>

Date/Publication 2026-08-08 13:40:45 UTC

RemoteUrl <https://github.com/kzst/nda>

RemoteRef HEAD

RemoteSha 1b692c2a332d1371fea79134629912b09f1166d7

Contents

nda-package	2
biplot.nda	3
COVID19_2020	4
CrimesUSA1990.X	4
CrimesUSA1990.Y	5
CWTS_2020	5
data_gen	6
dCor	7
dCov	7
fitted.ndrlm	8
fs.dimred	9
fs.KMO	9
GOVDB2020	10
I40_2020	11
ndr	11
ndrlm	14
normalize	16
pdCor	17
plot.nda	17
plot.ndrlm	19
predict.nda	20
predict.ndrlm	20
print.nda	21
print.ndrlm	22
residuals.ndrlm	22
spdCor	23
summary.nda	24
summary.ndrlm	24
Index	26

nda-package	<i>Package of Generalized Network-based Dimensionality Reduction and Analyses</i>
-------------	---

Description

The package of Generalized Network-based Dimensionality Reduction and Analysis (GNDA).

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona, Zahid Khan
e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyn, Z. T., Kurbucz, M. T., & Katona, A. I. (2022). Network-based dimensionality reduction of high-dimensional, low-sample-size datasets. Knowledge-Based Systems, 109180.

Kosztyn, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: <https://doi.org/10.1016/j.eswa.2023.121779>>

See Also

[ndr](#), [ndr1m](#), [plot](#), [biplot](#), [summary](#), [dCor](#).

biplot.nda	<i>Biplot NDA Scores and Loadings</i>
------------	---------------------------------------

Description

Creates a printable two-dimensional biplot or a sequence of three-factor projections.

Usage

```
## S3 method for class 'nda'
biplot(x, main = NULL, dims = c(1, 2), view_3D = FALSE,
       triples = NULL, interactive = FALSE, angle = 45,
       point_cex = 0.7, label_cex = 0.9, ...)
```

Arguments

- x An nda object fitted to row-level data.
- main Optional title.
- dims Two factors for the two-dimensional view.
- view_3D Display three-factor projections.
- triples Optional list of factor triples. By default all triples are used.
- interactive Use **plotly** for rotatable 3D views.
- angle Static axonometric projection angle in degrees.
- point_cex, label_cex Observation and loading-label size.
- ... Additional two-dimensional base-graphics arguments.

Value

An nda_biplot object; printing it renders the selected view or views.

Examples

```
fit <- ndr(swiss)
b <- biplot(fit, dims = c(1, 2))
b
```

COVID19_2020

COVID-19 case data for 138 countries and 18 variables

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)
 Covid' 19 of countries (2020), where the data frame has 138 observations of 18 variables.

Usage

```
data("COVID19_2020")
```

Format

A data frame with 138 observations 18 variables.

Source

Kurbucz, M. T. (2020). A joint dataset of official COVID-19 reports and the governance, trade and competitiveness indicators of World Bank group platforms. Data in brief, 31, 105881.

Examples

```
data(COVID19_2020)
```

CrimesUSA1990.X

Crimes in USA cities in 1990. Independent variables (X)

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)
 Crimes in USA cities in 1990. Independent variables (X)

Usage

```
data("CrimesUSA1990.X")
```

Format

A data frame with 1994 observations 123 variables.

Source

UCI - Machine Learning Repository: <https://archive.ics.uci.edu/ml/datasets/communities+and+crime>

Examples

```
data(CrimesUSA1990.X)
```

CrimesUSA1990.Y	<i>Crimes in USA cities in 1990. Dependent variable (Y)</i>
-----------------	---

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)
Crimes in USA cities in 1990. Dependent variable (Y)

Usage

```
data("CrimesUSA1990.Y")
```

Format

A data frame with 1994 observations 1 variables.

Source

UCI - Machine Learning Repository: <https://archive.ics.uci.edu/ml/datasets/communities+and+crime>

Examples

```
data(CrimesUSA1990.Y)
```

CWTS_2020	<i>CWTS Leiden University Ranking 2020 with 1,176 universities and 42 indicators</i>
-----------	--

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)
CWTS Leiden's 2020 dataset, where the data frame has 1176 observations of 42 variables.

Usage

```
data("CWTS_2020")
```

Format

A data frame with 1176 observations of 42 variables.

Source

CWTS Leiden Ranking 2020: <https://www.leidenranking.com/ranking/2020/list>

Examples

```
data(CWTS_2020)
```

data_gen	<i>Generate Structured NDA Example Data</i>
----------	---

Description

Generates a noisy block matrix without constructing intermediate block-diagonal matrices.

Usage

```
data_gen(n, m, nfactors = 2, lambda = 1, sparse = FALSE,
         density = 1, seed = NULL)
```

Arguments

n	Positive number of rows.
m	Positive number of columns.
nfactors	Positive number of blocks.
lambda	Finite exponential smoothing parameter.
sparse	Return a sparse Matrix object.
density	Proportion in (0, 1] of block entries retained as nonzero.
seed	Optional random seed.

Value

A data frame, or a sparse matrix when `sparse = TRUE`.

Examples

```
dense <- data_gen(30, 10, nfactors = 2, seed = 1)
sparse <- data_gen(100, 40, nfactors = 4, sparse = TRUE, density = 0.5)
```

dCor	<i>Distance-Correlation Matrix</i>
------	------------------------------------

Description

Calculates distance correlation for two vectors or for every unique pair of matrix columns.

Usage

```
dCor(x, y = NULL, test = FALSE, adjust = "BH", alpha = NULL,
      parallel = FALSE, cores = 1L, R = 499L)
```

Arguments

x	Numeric vector, matrix, or data frame.
y	Optional numeric vector paired with x.
test	Run permutation tests and return adjusted p-values.
adjust	Method passed to p.adjust.
alpha	Optional adjusted-p threshold used to zero nonsignificant coefficients.
parallel	Calculate unique matrix pairs in parallel on supported systems.
cores	Requested core count.
R	Permutation count passed to energy::dcor.test.

Value

A coefficient or symmetric matrix when test = FALSE; otherwise a list with r and p.

Examples

```
x <- matrix(rnorm(80), 20, 4)
dCor(x)
## Not run: dCor(x, test = TRUE, R = 999)
```

dCov	<i>Distance-Covariance Matrix</i>
------	-----------------------------------

Description

Calculates distance covariance for two vectors or for every unique pair of matrix columns.

Usage

```
dCov(x, y = NULL, test = FALSE, adjust = "BH", alpha = NULL,
      parallel = FALSE, cores = 1L, R = 499L)
```

Arguments

x	Numeric vector, matrix, or data frame.
y	Optional numeric vector paired with x.
test	Run permutation tests and return adjusted p-values.
adjust	Method passed to p.adjust.
alpha	Optional adjusted-p threshold used to zero nonsignificant off-diagonal values.
parallel	Calculate unique matrix pairs in parallel on supported systems.
cores	Requested core count.
R	Permutation count passed to energy::dcov.test.

Value

A covariance value or symmetric matrix when test = FALSE; otherwise a list with r and p.

Examples

```
x <- matrix(rnorm(80), 20, 4); dCov(x)
```

fitted.ndrlm	<i>NDRLM Fitted Values</i>
--------------	----------------------------

Description

Extracts fitted values from every response model.

Usage

```
## S3 method for class 'ndrlm'
fitted(object, ...)
```

Arguments

object	An ndrlm object.
...	Reserved for method compatibility.

Value

A data frame with one column per fitted response.

Examples

```
X <- as.data.frame(freeny.x)
Y <- data.frame(y = freeny.y)
fit <- ndrlm(Y, X, optimize = FALSE)
fitted(fit)
```

fs.dimred	<i>Loading-Based Feature Selection</i>
-----------	--

Description

Selects indicators from NDA, PCA, factor-analysis, and other objects that expose loadings, rotations, or variable coordinates.

Usage

```
fs.dimred(fn, DF, min_comm = 0.25, com_comm = 0.25, ...)
```

Arguments

fn	Fitted dimensionality-reduction object. Supported structures include nda, prcomp, princomp, psych PCA/FA, and objects with loadings, rotation, or var\$coord.
DF	Original numeric data or a symmetric correlation matrix.
min_comm	Minimum indicator communality.
com_comm	Minimum separation between the two largest squared loadings.
...	Reserved for future model-specific extraction.

Value

A list containing dropped_low, dropped_com, retained_DF (also available as remain_DF), communalities, loadings, and the input model.

Examples

```
pca <- prcomp(swiss, scale. = TRUE)
selected <- fs.dimred(pca, swiss)
colnames(selected$retained_DF)
```

fs.KMO	<i>KMO-Based Feature Selection</i>
--------	------------------------------------

Description

Iteratively removes the indicator with the lowest measure of sampling adequacy until the requested overall KMO is reached.

Usage

```
fs.KMO(data, min_MSA = 0.5, cor.mtx = FALSE,
        max_iter = ncol(data) - 1L, ...)
```

Arguments

<code>data</code>	Numeric row-level data or a correlation matrix.
<code>min_MSA</code>	Target overall Kaiser-Meyer-Olkin value in $[0, 1]$.
<code>cor.mtx</code>	Treat data as a symmetric correlation matrix.
<code>max_iter</code>	Maximum number of removals.
<code>...</code>	Reserved for backward compatibility.

Value

Filtered data or correlation matrix. The final KMO result and removed variable names are stored in "KMO" and "removed" attributes.

Examples

```
filtered <- fs.KMO(swiss, min_MSA = 0.6)
```

GOVDB2020

Governmental and economic data for 138 countries and 2,161 variables

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)
 Governmental and economic data of countries (2020), where the data frame has 138 observations of 2161 variables.

Usage

```
data("GOVDB2020")
```

Format

A data frame with 138 observations of 2161 variables.

Source

Kurbucz, M. T. (2020). A joint dataset of official COVID-19 reports and the governance, trade and competitiveness indicators of World Bank group platforms. Data in brief, 31, 105881.

Examples

```
data(GOVDB2020)
```

I40_2020	<i>NUTS2 regional Industry 4.0 readiness data with 414 observations and 101 variables</i>
----------	---

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA) NUTS2 regional development data (2020), where the data frame has 414 observations of 101 variables.

Usage

```
data("I40_2020")
```

Format

A data frame with 414 observations of 101 variables.

Source

Honti, G., Czvetko, T., and Abonyi, J. (2020). Data describing the regional Industry 4.0 readiness index. Data in Brief, 33, 106464.

Examples

```
data(I40_2020)
```

ndr	<i>Generalized Network-Based Dimensionality Reduction and Analysis</i>
-----	--

Description

Builds a squared-association network, detects indicator communities, and estimates one centrality-weighted latent variable per retained community.

Usage

```
ndr(r, covar = FALSE, cor_method = 1, cor_type = 1, min_R = 0,
    min_comm = 2, Gamma = 1, null_model_type = 4, mod_mode = 6,
    min_evalue = 0, min_communality = 0, com_communalities = 0,
    use_rotation = FALSE, rotation = "oblimin", weight = NULL,
    seed = NULL, parallel = FALSE, cores = 1L, sparsecalc = FALSE,
    membership = NULL,
    centrality = c("eigenvector", "authority", "hub", "pagerank"),
    standardized = TRUE, P_mat = NULL, test = FALSE,
    p_adjust = "BH", alpha = NULL, bootstrap = FALSE,
    n_boot = 200L, conf = 0.95)
```

Arguments

<code>r</code>	Numeric row-level data, or a square association matrix when <code>covar = TRUE</code> . Sparse Matrix inputs are accepted.
<code>covar</code>	Use <code>r</code> as an association matrix instead of estimating associations from rows.
<code>cor_method</code>	Association method. Numeric values 1–6 retain compatibility; textual values are "pearson", "spearman", "kendall", "distance", "gaussian_rank", and "biweight". Gaussian-rank correlation is a fast monotone robust alternative; biweight correlation limits outlier influence.
<code>cor_type</code>	Association type: "full", "partial", or "semipartial"; 1–3 are accepted aliases.
<code>min_R</code>	Minimum squared association retained as an edge.
<code>min_comm</code>	Minimum preferred community size. A singleton is retained when it is the only feasible result.
<code>Gamma</code>	Positive resolution multiplier used by community detection and null-model subtraction.
<code>null_model_type</code>	Null model: 1 configuration-model expectation, 2 mean positive similarity, 3 <code>min_R</code> , or 4 no matrix modification (default).
<code>mod_mode</code>	Community method: Louvain, fast greedy, leading eigenvector, Infomap, Walktrap, or Leiden; either names or values 1–6.
<code>min_evalue</code>	Minimum selected centrality weight for periphery filtering.
<code>min_communality</code>	Minimum indicator communality.
<code>com_communalities</code>	Minimum separation between the two largest squared loadings for common-indicator filtering. Zero disables this filter.
<code>use_rotation</code>	Rotate a multi-factor solution.
<code>rotation</code>	Rotation understood by <code>psych::principal</code> .
<code>weight</code>	One finite observation-score weight per indicator.
<code>seed</code>	Optional random seed.
<code>parallel</code>	Use multicore calculation for pairwise distance associations and bootstrap replicates where supported.
<code>cores</code>	Number of requested cores.
<code>sparsecalc</code>	Store and process the thresholded similarity network as a sparse matrix.
<code>membership</code>	Optional supplied community membership, one value per indicator. Zero excludes an indicator.
<code>centrality</code>	Latent-variable weighting: eigenvector, authority, hub, or PageRank centrality.
<code>standardized</code>	Standardize input indicators using training means and standard deviations before weighting.
<code>P_mat</code>	Optional finite expected-edge matrix. When supplied it overrides <code>null_model_type</code> .
<code>test</code>	Calculate association p-values.

p_adjust	Method passed to p.adjust for false-discovery-rate control.
alpha	Optional adjusted-p threshold; nonsignificant associations are set to zero.
bootstrap	Bootstrap loading intervals from row-level data.
n_boot	Number of bootstrap replicates.
conf	Bootstrap confidence level.

Details

The method follows three stages: construct a squared-association graph, detect network communities, and form standardized centrality-weighted latent scores. Partial and semipartial networks can be asymmetric. Null-model types 1–3 operate on nonnegative residual similarities; type 4 leaves the similarity matrix unchanged.

Value

An object of class `nda`. Important elements include scores, loadings, communality, uniqueness, membership, EVCs, R, correlation, modularity, reconstruction, and stats. The stats list contains indicator p-values, optional bootstrap intervals, retained and dropped indicators, community sizes, and null-model metadata. Scores are unavailable for association-matrix input, but reconstruction-consistent loadings, communalities, and a loading-based reconstructed association matrix are returned.

References

Kosztyn, Z. T., Kurbucz, M. T., and Katona, A. I. (2022). Network-based dimensionality reduction of high-dimensional, low-sample-size datasets. *Knowledge-Based Systems*, 109180. doi:10.1016/j.knosys.2022.109180.

Kosztyn, Z. T., Katona, A. I., Kurbucz, M. T., and Lantos, Z. (2024). Generalized network-based dimensionality analysis. *Expert Systems with Applications*, 238, 121779. doi:10.1016/j.eswa.2023.121779.

Examples

```
fit <- ndr(swiss, cor_method = "spearman", centrality = "pagerank")
summary(fit)
scores <- predict(fit, swiss[1:3, ])

association <- cor(swiss)
fit_matrix <- ndr(association, covar = TRUE, membership = c(1, 1, 1, 2, 2, 2))
fit_matrix$loadings

## Not run:
fit_boot <- ndr(swiss, cor_method = "distance", test = TRUE,
               bootstrap = TRUE, n_boot = 500, parallel = TRUE, cores = 4)

## End(Not run)
```

ndr1m

Network-Based Dimensionality Reduction and Regression

Description

Fits supervised models to original or NDA-reduced input and output variables, with optional NSGA-II tuning.

Usage

```
ndr1m(Y, X, latents = "in", dircon = FALSE, optimize = TRUE,
      target = "adj.r.square", rel_weight = FALSE, cor_method = 1,
      cor_type = 1, min_comm = 2, Gamma = 1, null_model_type = 4,
      mod_mode = 1, use_rotation = FALSE, rotation = "oblimin",
      pareto = FALSE, fit_weights = NULL,
      lower.bounds.x = rep(-100, ncol(X)),
      upper.bounds.x = rep(100, ncol(X)),
      lower.bounds.latentx = c(0, 0, 0, 0),
      upper.bounds.latentx = c(0.6, 0.6, 0.6, 0.3),
      lower.bounds.y = rep(-100, ncol(Y)),
      upper.bounds.y = rep(100, ncol(Y)),
      lower.bounds.latenty = c(0, 0, 0, 0),
      upper.bounds.latenty = c(0.6, 0.6, 0.6, 0.3),
      popsize = 20, generations = 30, cprob = 0.7, cdist = 5,
      mprob = 0.2, mdist = 10, seed = NULL,
      regression = c("lm", "ridge", "loess", "elastic_net"),
      model = NULL, alpha = 0.5, lambda = NULL, loess_span = 0.75,
      restarts = 1L, parallel = FALSE, cores = 1L,
      sparsecalc = FALSE, centrality = "eigenvector",
      standardized = TRUE, membership.X = NULL, membership.Y = NULL)
```

Arguments

Y	Numeric dependent-variable data.
X	Numeric independent-variable data with the same rows as Y.
latents	Use latent variables for inputs ("in"), outputs ("out"), both, or neither.
dircon	Retain NDA-excluded variables as direct predictors or responses.
optimize	Tune indicator weights and four NDA thresholds with NSGA-II.
target	One of adjusted R-squared, R-squared, MAE, MAPE, MASE, MSE, RMSE, AIC, or BIC.
rel_weight	Constrain and normalize indicator weights to be nonnegative relative weights.
cor_method, cor_type, min_comm, Gamma, null_model_type, mod_mode, use_rotation, rotation	Arguments forwarded to ndr.

<code>pareto</code>	Keep one objective per original dependent variable during NSGA-II optimization.
<code>fit_weights</code>	Optional weights for aggregating response-specific objectives when their number matches the fitted responses.
<code>lower.bounds.x, upper.bounds.x</code>	Lower and upper NSGA-II bounds for input-indicator weights.
<code>lower.bounds.latentx, upper.bounds.latentx</code>	Bounds for input-side <code>min_evalue</code> , <code>min_communality</code> , <code>com_communalities</code> , and <code>min_R</code> .
<code>lower.bounds.y, upper.bounds.y</code>	Lower and upper NSGA-II bounds for output-indicator weights.
<code>lower.bounds.latency, upper.bounds.latency</code>	Bounds for the four output-side NDA thresholds.
<code>popsize, generations, cprob, cdist, mprob, mdist</code>	NSGA-II population, generation, crossover, and mutation settings passed to <code>mco::nsga2</code> .
<code>seed</code>	Optional seed; repeated optimization runs use consecutive seeds.
<code>regression</code>	Regression engine: ordinary least squares, ridge, LOESS, or elastic net.
<code>model</code>	Backward-friendly alias that overrides <code>regression</code> .
<code>alpha</code>	Elastic-net mixing value in $[0, 1]$. Ridge always uses zero.
<code>lambda</code>	Optional penalization value. If omitted, cross-validation selects <code>lambda</code> .
<code>loess_span</code>	LOESS smoothing span.
<code>restarts</code>	Number of independent NSGA-II runs; the best feasible candidate is refitted.
<code>parallel, cores, sparsecalc, centrality, standardized</code>	Performance and latent-score settings forwarded to <code>ndr</code> .
<code>membership.X, membership.Y</code>	Optional fixed NDA community memberships for the input and output sides.

Details

NDRLM is predictive and exploratory; its paths do not by themselves establish causal effects. Ridge and elastic net use **glmnet**. LOESS is intended for small predictor sets and does not provide linear-path significance tests. AIC and BIC for non-likelihood engines are residual-based approximations.

Value

An `ndrlm` object containing fitted regression models, response-specific metrics, fitted NDA objects, optimized hyperparameters, training designs, and NSGA-II results.

References

Kosztyn, Z. T., Kurbucz, M. T., Katona, A. I., and Khan, Z. Network-based dimensionality reduction and structural modeling.

Examples

```

X <- as.data.frame(freeny.x)
Y <- data.frame(income = freeny.y)
fit <- ndrmlm(Y, X, optimize = FALSE, regression = "lm")
summary(fit)
predict(fit, X[1:3, ])

## Not run:
fit_en <- ndrmlm(Y, X, regression = "elastic_net", alpha = 0.5,
                 popsize = 20, generations = 50, restarts = 3)

## End(Not run)

```

normalize

Fast Min-Max Normalization

Description

Rescales a numeric matrix or data frame globally, by row, or by column.

Usage

```

normalize(x, type = c("all", "row", "col"), range = c(0, 1),
         na.rm = TRUE)

```

Arguments

x	Numeric matrix or data frame.
type	Normalize all values together, each row, or each column.
range	Two increasing finite output limits.
na.rm	Remove missing values when finding limits.

Details

A constant slice is mapped to the midpoint of range.

Value

A normalized object with the same matrix or data-frame form as x.

Examples

```

m <- matrix(rnorm(20), 5, 4)
normalize(m)
normalize(m, type = "row", range = c(-1, 1))

```

pdCor

*Partial Distance-Correlation Matrix***Description**

Forms a partial distance-correlation matrix by inverting the distance-covariance matrix.

Usage

```
pdCor(x, test = FALSE, adjust = "BH", alpha = NULL,
      parallel = FALSE, cores = 1L)
```

Arguments

x	Numeric matrix or data frame.
test	Return approximate coefficient p-values.
adjust	Method passed to p.adjust.
alpha	Optional adjusted-p filtering threshold.
parallel	Calculate distance-covariance pairs in parallel where supported.
cores	Requested core count.

Details

When the distance-covariance matrix is singular, a Moore-Penrose generalized inverse is used. The optional p-values use a t approximation and should be interpreted as screening diagnostics.

Value

A matrix, or a list with matrices r and p when test = TRUE.

Examples

```
x <- matrix(rnorm(100), 20, 5); pdCor(x)
```

plot.nda

*Plot an NDA Association Network***Description**

Creates a printable static **igraph** or interactive **visNetwork** network object. Assignment does not draw; printing the result draws it.

Usage

```
## S3 method for class 'nda'
plot(x, cuts = 0.3, interactive = TRUE, edgescale = 1,
      labeldist = -1.5, show_weights = FALSE, shapes = "circle",
      edge.color.links = TRUE,
      edge.colors = c("darkred", "grey", "darkblue"),
      edge.label.size = 10, edge.label.color = "#333333",
      vertex.label.size = 12, vertex.label.color = "black",
      layout = "nicely", igraph_args = list(), visnetwork_args = list(), ...)
```

Arguments

x	An nda object.
cuts	Minimum squared correlation, or a two-value retained range. For association-matrix models a scalar is interpreted relative to the nonzero minimum and maximum.
interactive	Create a visNetwork widget instead of a static graph.
edgescale	Edge-width multiplier.
labeldist	Static vertex-label distance.
show_weights	Label edges with unsquared correlations for row-level models and supplied similarities for association-matrix models.
shapes	One or more vertex shapes, recycled over indicators.
edge.color.links	Color links by signed association or by the association-matrix minimum, median, and maximum.
edge.colors	Three-color low/middle/high scale.
edge.label.size, edge.label.color	Edge-label appearance.
vertex.label.size, vertex.label.color	Vertex-label appearance.
layout	An igraph layout suffix such as "nicely", "fr", or "circle".
igraph_args	Named static plot arguments.
visnetwork_args	Named lists called network, nodes, edges, options, and layout, forwarded to matching visNetwork functions.
...	Additional static plot.igraph arguments.

Value

An object of class `nda_plot`; print it to render.

Examples

```
fit <- ndr(swiss)
p <- plot(fit, interactive = FALSE, show_weights = TRUE)
p
```

plot.ndrlm	<i>Plot an NDRLM Path Network</i>
------------	-----------------------------------

Description

Creates a layered, printable network of indicators, latent variables, and fitted regression links.

Usage

```
## S3 method for class 'ndrlm'
plot(x, sig = 0.05, interactive = FALSE,
     show_edge_labels = TRUE, edge.label.size = 9,
     vertex.label.size = 11, edgescale = 2,
     igraph_args = list(), visnetwork_args = list(), ...)
```

Arguments

x	An ndrlm object.
sig	Significance threshold for ordinary least-squares paths. Penalized and LOESS paths do not have p-values and are not filtered by sig.
interactive	Create a visNetwork widget.
show_edge_labels	Show loading and path estimates.
edge.label.size, vertex.label.size	Label sizes.
edgescale	Absolute-weight edge-width multiplier.
igraph_args	Named static plotting arguments.
visnetwork_args	Named network, options, and physics argument lists.
...	Additional static plot.igraph arguments.

Value

An ndrlm_plot object; printing it renders the graph.

Examples

```
X <- as.data.frame(freeny.x)
Y <- data.frame(y = freeny.y)
p <- plot(ndrlm(Y, X, optimize = FALSE))
p
```

predict.nda

Predict NDA Scores

Description

Applies training feature standardization, indicator weights, retained centralities, latent-score scaling, and rotation to new rows.

Usage

```
## S3 method for class 'nda'
predict(object, newdata, ...)
```

Arguments

object	An nda object fitted to row-level data.
newdata	Numeric data containing every original training indicator.
...	Reserved for method compatibility.

Value

A matrix with one predicted score column per latent variable. If newdata is NULL, training scores are returned.

Examples

```
fit <- ndr(swiss); predict(fit, swiss[1:4, ])
```

predict.ndrlm

Predict from an NDRLM Model

Description

Transforms new independent variables with the fitted NDA model and predicts every fitted response.

Usage

```
## S3 method for class 'ndrlm'
predict(object, newdata = NULL, se.fit = FALSE,
        interval = c("none", "confidence", "prediction"), level = 0.95, ...)
```

Arguments

<code>object</code>	An <code>ndr1m</code> object.
<code>newdata</code>	New original independent variables. If omitted, fitted values are returned.
<code>se.fit</code>	Request standard errors for ordinary least-squares fits.
<code>interval</code>	No interval, a confidence interval, or a prediction interval. Intervals are available for ordinary least squares only.
<code>level</code>	Confidence level.
<code>...</code>	Additional arguments passed to <code>predict.lm</code> .

Value

A response-named data frame for ordinary predictions, or a response-named list when standard errors or intervals are requested.

Examples

```
X <- as.data.frame(freeny.x); Y <- data.frame(income = freeny.y)
fit <- ndr1m(Y, X, optimize = FALSE)
predict(fit, X[1:3, ])
```

print.nda

Print an NDA Model

Description

Prints the NDA call, settings, modularity, community sizes, loadings, communalities, and dropped indicators.

Usage

```
## S3 method for class 'nda'
print(x, digits = getOption("digits"), ...)
```

Arguments

<code>x</code>	An <code>nda</code> object.
<code>digits</code>	Printing precision.
<code>...</code>	Additional arguments passed to the summary printer.

Value

`x`, invisibly.

Examples

```
fit <- ndr(swiss); print(fit)
```

print.ndrlm	<i>Print an NDRLM Model</i>
-------------	-----------------------------

Description

Prints the reduction mode, regression engine, target metrics, and response-specific fits.

Usage

```
## S3 method for class 'ndrlm'
print(x, digits = getOption("digits"), ...)
```

Arguments

x	An ndrlm object.
digits	Printing precision.
...	Additional arguments passed to the summary printer.

Value

x, invisibly.

Examples

```
X <- as.data.frame(freeny.x)
Y <- data.frame(y = freeny.y)
print(ndrlm(Y, X, optimize = FALSE))
```

residuals.ndrlm	<i>NDRLM Residuals</i>
-----------------	------------------------

Description

Extracts observed-minus-fitted residuals from every response model.

Usage

```
## S3 method for class 'ndrlm'
residuals(object, ...)
```

Arguments

object	An ndrlm object.
...	Reserved for method compatibility.

Value

A data frame with one residual column per fitted response.

Examples

```
X <- as.data.frame(freeny.x)
Y <- data.frame(y = freeny.y)
fit <- ndrlm(Y, X, optimize = FALSE)
residuals(fit)
```

 spdCor

Semipartial Distance-Correlation Matrix

Description

Forms an asymmetric semipartial distance-correlation matrix from distance covariances.

Usage

```
spdCor(x, test = FALSE, adjust = "BH", alpha = NULL,
       parallel = FALSE, cores = 1L)
```

Arguments

x	Numeric matrix or data frame.
test	Return approximate coefficient p-values.
adjust	Method passed to p.adjust.
alpha	Optional adjusted-p filtering threshold.
parallel	Calculate distance-covariance pairs in parallel where supported.
cores	Requested core count.

Details

When the distance-covariance matrix is singular, a Moore-Penrose generalized inverse is used. The optional p-values use a t approximation and should be interpreted as screening diagnostics.

Value

A matrix, or a list with matrices r and p when test = TRUE.

Examples

```
x <- matrix(rnorm(100), 20, 5); spdCor(x)
```

summary.nda

Summarize an NDA Model

Description

Returns a printable summary without producing output during assignment.

Usage

```
## S3 method for class 'nda'
summary(object, digits = getOption("digits"), ...)
```

Arguments

object	An nda object.
digits	Preferred printing precision.
...	Additional summary arguments.

Value

An object of class summary.nda containing settings, modularity, community sizes, loadings, communalities, uniquenesses, and inference statistics.

Examples

```
fit <- ndr(swiss); s <- summary(fit); s
```

summary.ndrlm

Summarize an NDRLM Model

Description

Returns a printable regression and reduction summary without printing during assignment.

Usage

```
## S3 method for class 'ndrlm'
summary(object, digits = getOption("digits"), ...)
```

Arguments

object	An ndrlm object.
digits	Printing precision.
...	Additional summary arguments.

Value

A `summary.ndrlm` object containing settings, variables, response metrics, fitted NDA objects, and regression fits.

Examples

```
X <- as.data.frame(freeny.x)
Y <- data.frame(y = freeny.y)
s <- summary(ndrlm(Y, X, optimize = FALSE))
s
```

Index

- * **array**
 - data_gen, 6
- * **datasets**
 - COVID19_2020, 4
 - CrimesUSA1990.X, 4
 - CrimesUSA1990.Y, 5
 - CWTS_2020, 5
 - GOVDB2020, 10
 - I40_2020, 11
- * **matrix**
 - normalize, 16
- * **multivariate**
 - dCor, 7
 - dCov, 7
 - fitted.ndrml, 8
 - fs.dimred, 9
 - fs.KM0, 9
 - ndr, 11
 - ndrml, 14
 - pdCor, 17
 - plot.nda, 17
 - plot.ndrml, 19
 - predict.nda, 20
 - predict.ndrml, 20
 - print.nda, 21
 - print.ndrml, 22
 - residuals.ndrml, 22
 - spdCor, 23
 - summary.nda, 24
 - summary.ndrml, 24
- * **package**
 - nda-package, 2
- * **plot**
 - biplot.nda, 3
- biplot, 3
- biplot.nda, 3
- COVID19_2020, 4
- CrimesUSA1990.X, 4
- CrimesUSA1990.Y, 5
- CWTS_2020, 5
- data_gen, 6
- dCor, 3, 7
- dCov, 7
- fitted.ndrml, 8
- fs.dimred, 9
- fs.KM0, 9
- GOVDB2020, 10
- I40_2020, 11
- nda (nda-package), 2
- nda-package, 2
- ndr, 3, 11
- ndrml, 3, 14
- normalize, 16
- pdCor, 17
- plot, 3
- plot.nda, 17
- plot.ndrml, 19
- predict.nda, 20
- predict.ndrml, 20
- print.nda, 21
- print.ndrml, 22
- residuals.ndrml, 22
- spdCor, 23
- summary, 3
- summary.nda, 24
- summary.ndrml, 24