Package ‘netmem’

April 16, 2026

Version 1.0-3

Date 2026-04-01

Depends R (>= 4.0.0)

Imports igraph, Matrix, stats

Suggests knitr, rmarkdown, covr, testthat, usethis, styler

Title Social Network Measures using Matrices

Maintainer Alejandro Espinosa-Rada <anespinosa@uc.cl>

Description Provides measures to describe and manipulate one-mode, two-mode, multiplex, and multilevel networks using matrix algebra. Implements functions for network centrality, cohesive subgroups, structural holes, similarity measures, path distances, signed networks, and random network generation. Supports ego-centric and whole-network analyses, including dyadic and triadic census, structural balance, and bipartite projections.
Key references: Bonacich (1972) <doi:10.1080/0022250X.1972.9989806>,
Breiger (1974) <doi:10.2307/2576011>,
Kivelä et al. (2014) <doi:10.1093/comnet/cnu016>,
Espinosa-Rada et al. (2024) <doi:10.1016/j.socnet.2023.11.008>.

License GPL-3

Encoding UTF-8

Language en-US

URL https://github.com/anespinosa/netmem,
https://anespinosa.github.io/netmem/

BugReports https://github.com/anespinosa/netmem/issues

VignetteBuilder knitr

RoxygenNote 7.3.3

LazyData true

NeedsCompilation no

Author Alejandro Espinosa-Rada [cre, aut] (ORCID:
<https://orcid.org/0000-0003-4177-1912>)

Repository CRAN

Date/Publication 2026-04-16 18:20:11 UTC

Contents

`adj_to_incidence` *Convert an Adjacency Matrix to an Incidence Matrix*

Description

This function transforms an adjacency matrix into an incidence matrix.

Usage

```
adj_to_incidence(A, loops = TRUE, directed = TRUE, weighted = TRUE)
```

Arguments

`A`	A square numeric matrix representing the adjacency matrix of a graph. The matrix should have non-negative values, where ‘A[i, j]‘ represents the weight of the edge from node ‘i‘ to node ‘j‘.
`loops`	Logical. If ‘TRUE‘, self-loops (edges from a node to itself) are included in the incidence matrix. If ‘FALSE‘, they are removed. Default is ‘TRUE‘.
`directed`	Logical. If ‘TRUE‘, the graph is treated as directed, meaning each edge has a specific source and target. If ‘FALSE‘, the graph is treated as undirected, and edges are symmetrically represented. Default is ‘TRUE‘.
`weighted`	Logical. If ‘TRUE‘, edge weights from ‘A‘ are included in the incidence matrix. If ‘FALSE‘, all edges are treated as having weight ‘1‘. Default is ‘TRUE‘.

Value

A numeric matrix where rows represent nodes and columns represent edges. - In a **directed** network, a source node has a negative value (-weight), and a target node has a positive value (+weight). - In an **undirected** network, both nodes involved in an edge share the weight (positive values). - If ‘weighted = FALSE‘, all edges have a weight of ‘1‘.

Examples

```
# Define an adjacency matrix (directed and weighted)
A <- matrix(c(
  1, 3, 0, 0, 2,
  0, 0, 2, 0, 0,
  5, 0, 0, 0, 0,
  0, 0, 0, 0, 1,
  0, 4, 0, 0, 0
), byrow = TRUE, nrow = 5)

# Convert to an incidence matrix (directed, weighted)
(inc_matrix <- adj_to_incidence(A))

# Undirected, weighted graph
(inc_matrix_undirected <- adj_to_incidence(A, directed = FALSE))

# Directed, unweighted graph
(inc_matrix_unweighted <- adj_to_incidence(A, weighted = FALSE))

# Ignore loops
(inc_matrix_no_loops <- adj_to_incidence(A, loops = FALSE))
```

---

adj_to_matrix *Transform an adjacency list into a matrix*

Description

Transform an adjacency list into a matrix

Usage

```
adj_to_matrix(A, type = c("adjacency", "incidence", "weighted"), loops = FALSE)
```

Arguments

A	An adjacent list
type	Transform the adjacent list into an `adjacency` matrix, an `incidence` matrix or a `weighted` matrix
loops	Whether to include loops into the matrix

Value

This function transforms an adjacency list into a matrix

Author(s)

Alejandro Espinosa-Rada

Examples

```
adj_groups <- rbind(
  c("a", "b", "c"), c("a", "c", NA),
  c("b", "c", NA), c("c", NA, NA),
  c("c", "a", NA)
)
M <- adj_to_matrix(adj_groups, type = "adjacency", loops = TRUE)
M
```

---

bonacich_norm *Bonacich normalization*

Description

The function provide a normalisation provided by Bonacich (1972).

Usage

```
bonacich_norm(A, projection = c("rows", "columns"), normalisation = FALSE)
```

Arguments

A	An incidence matrix
projection	Whether to normalise by rows (default), or columns of the matrix.
normalisation	Normalise the measure

Value

This function returns the Bonacich normalisation.

Source

Adapted from Borgatti, S., Everett, M., Johnson, J. and Agneessens, P. (2022) Analyzing Social Networks Using R. Sage.

References

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. Journal of Mathematical Sociology, 2: 112-120.

Examples

```
A <- matrix(
  c(
    1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0,
    1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
    1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
    0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1,
    0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
    0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,
    0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0
  ),
  byrow = TRUE, ncol = 14
)
bonacich_norm(A)
```

---

clique_table *Clique table*

Description

Exploration of a 3-cliques, as the maximum number of three or more actors who have all possible ties present among themselves

Usage

```
clique_table(A, list_cliques = FALSE, number = FALSE)
```

Arguments

A	A symmetric matrix object.
list_cliques	Whether to return the list of cliques.
number	Number of triangles

Value

This function return an edge list of actors participating in 3-cliques.

If `list_cliques = TRUE` it also return the list of cliques per nodes. If `number = TRUE` the output returns the number of 3-cliques in the matrix.

Author(s)

Alejandro Espinosa-Rada

References

Luce, R.D. and Perry, A.D. (1949). A method of matrix analysis of group structure. Psychometrika, 14: 95-116.

Roethlisberger, F.J. and Dickson, W.J. (1939). Management and the Worker. Harvard University Press, Cambridge, MA.

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  0, 1, 1, 0, 0, 0, 0, 1, 0,
  1, 0, 1, 0, 0, 0, 0, 0, 0,
  1, 1, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 1, 1, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 1, 1, 0,
  0, 0, 0, 0, 0, 1, 0, 1, 0,
  1, 0, 0, 0, 0, 1, 1, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0
), byrow = TRUE, ncol = 9)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- rownames(A)
clique_table(A, list_cliques = TRUE, number = TRUE)
```

---

components_id *Components*

Description

This function assigns an id to the components that each of the nodes of the matrix belongs

Usage

```
components_id(A)
```

Arguments

A	A matrix

Value

A vector assigning an id the components that each of the nodes of the matrix belongs

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  0, 1, 1, 0, 0,
  1, 0, 1, 0, 0,
  1, 1, 0, 0, 0,
  0, 0, 0, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)
rownames(A) <- letters[1:ncol(A)]
colnames(A) <- rownames(A)
components_id(A)
```

---

compound_relation *Relational composition*

Description

This function returns the relational composition of the given matrices. The compound relations define the paths and the social process flows of the given matrices (Pattison, 1993). However, those whom they link may or may not be aware of them. The compound relations allow us to identify "the possibly very long and devious chains of effects propagating withing concrete social systems through links of various kinds" (Lorrain & White, 1971: 50).

Usage

```
compound_relation(l = list(), comp = 3, matrices = FALSE, equate = FALSE)
```

Arguments

`l`	A list of matrices.
`comp`	A number with the length of paths to form the compound relation.
`matrices`	Whether to return the resulting matrices of the compound relations.
`equate`	Whether to return the semigroup equations.

Value

This function provides the composition or concatenation of compound relations and the primitives of the matrices.

Author(s)

Alejandro Espinosa-Rada

References

Boorman, Scott A. and White, Harrison C. (1976) Social Structure from Multiple Networks. II. Role Structures. American Journal of Sociology. 81(6): 1384-1446.

Lorrain, Francois and White, Harrison C. (1971) Structural Equivalence of Individuals in Social Networks. Journal of Mathematical Sociology. 1: 49-80

Pattison, Philippa (1993) Algebraic Models for Social Networks. Cambridge University Press.

Examples

```
A <- matrix(c(
  0, 1, 0, 0,
  1, 0, 0, 0,
  1, 1, 0, 1,
  0, 0, 1, 0
), byrow = TRUE, ncol = 4)
rownames(A) <- letters[1:NCOL(A)]
colnames(A) <- rownames(A)

B <- matrix(c(
  0, 1, 0, 0,
  1, 0, 0, 0,
  0, 0, 0, 1,
  0, 0, 1, 0
), byrow = TRUE, ncol = 4)
rownames(B) <- letters[1:NCOL(B)]
colnames(B) <- rownames(B)

cmp <- compound_relation(list(A, B), comp = 2, matrices = TRUE, equate = TRUE)
cmp$compound_relations
cmp$compound_matrices
cmp$equated
```

---

co_occurrence *Co-occurrence*

Description

Co-occurrence matrix based on overlap function

Usage

```
co_occurrence(
  A,
  similarity = c("ochiai", "cosine"),
  occurrence = TRUE,
  projection = FALSE
)
```

Arguments

`A`	A matrix
`similarity`	The similarities available are either `Ochiai` (default) or `cosine`.
`occurrence`	Whether to treat the matrix as a two-mode structure (a.k.a. rectangular matrix, occurrence matrix, affiliation matrix, bipartite network)
`projection`	Whether to apply a projection (inner product multiplication) to the matrix

Value

This function returns the normalisation of a matrix into a symmetrical co-occurrence matrix

Author(s)

Alejandro Espinosa-Rada

References

Borgatti, S. P., Halgin, D. S., 2011. Analyzing affiliation networks. In: J. Scott and P. J. Carrington (Eds.) The Sage handbook of social network analysis (pp. 417-433), Sage.

Zhou, Q., & Leydesdorff, L. (2016). The normalization of occurrence and Co-occurrence matrices in bibliometrics using Cosine similarities and Ochiai coefficients. Journal of the Association for Information Science and Technology, 67(11), 2805–2814. doi:10.1002/asi.23603

Examples

```
A <- matrix(
  c(
    2, 0, 2,
    1, 1, 0,
    0, 3, 3,
    0, 2, 2,
    0, 0, 1
  ),
  nrow = 5, byrow = TRUE
)

co_occurrence(A)
```

`cumulativeSumMatrices` *Cumulative sum of matrices*

Description

Cumulative sum of matrices

Usage

```
cumulativeSumMatrices(matrixList)
```

Arguments

`matrixList`	A list of matrices

Value

This function returns the cumulative sum of matrices

Author(s)

Alejandro Espinosa-Rada

Examples

```
A <- matrix(c(
  0, 1, 1,
  0, 0, 0,
  0, 1, 0
), byrow = TRUE, ncol = 3)
B <- matrix(c(
  0, 0, 1,
  0, 0, 0,
  0, 0, 0
), byrow = TRUE, ncol = 3)
C <- matrix(c(
  0, 0, 0,
  1, 0, 0,
  0, 0, 0
), byrow = TRUE, ncol = 3)
matrixList <- list(A, B, C)
cumulativeSumMatrices(matrixList)
```

`distances` *Path distances*

Description

Distances between nodes using breadth-first search (BFS) or Dijkstra's algorithm to find shortest path distances.

Usage

```
bfs_ugraph(A, from = NULL)

count_geodesics(A)

short_path(A, from = NULL, to = NULL)

wlocal_distances(A, select = c("all", "in", "out"), from, to, path = c())

wall_distances(A, select = c("all", "in", "out"))
```

Arguments

`A`	A symmetric matrix object
`from`	Node in which the path start
`to`	Node in which the path end
`select`	Whether to consider all sender and receiver ties of ego (`all`), only incoming ties (`in`), or outgoing ties (`out`). By default, `all`.
`path`	Path of the nodes

Value

This function returns the distances o shortest path distance between two nodes for unweighted graph (`bfs_ugraph`, `count_geodesics` and `short_path` respectively) and weighted graphs (`wlocal_distances` or `wall_distances`)

Author(s)

Alejandro Espinosa-Rada

References

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik. 1: 269–271.

Examples

```
A <- matrix(c(
  0, 1, 1, 0, 0, 0,
  0, 0, 0, 1, 1, 0,
  0, 0, 0, 0, 1, 0,
  0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 1,
  0, 0, 0, 0, 0, 0
), byrow = TRUE, nrow = 6)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]

bfs_ugraph(A, from = "a")


A <- matrix(c(
  0, 1, 1, 0, 0, 0,
  0, 0, 0, 1, 1, 0,
  0, 0, 0, 0, 1, 0,
  0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 1,
  0, 0, 0, 0, 0, 0
), byrow = TRUE, nrow = 6)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]

count_geodesics(A)


A <- matrix(c(
  0, 1, 1, 0, 0, 0,
  0, 0, 0, 1, 1, 0,
  0, 0, 0, 0, 1, 0,
  0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 1,
  0, 0, 0, 0, 0, 0
), byrow = TRUE, nrow = 6)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]

short_path(A, from = "a", to = "d")


A <- matrix(
  c(
    0, 3, 3, 10, 15, 0, 0, 0,
    1, 0, 5, 2, 7, 0, 0, 0,
    3, 5, 0, 0, 0, 0, 0, 0,
    10, 2, 0, 0, 2, 7, 12, 0,
    11, 3, 0, 3, 0, 11, 2, 0,
    0, 0, 0, 7, 11, 0, 3, 2,
    0, 0, 0, 12, 2, 3, 0, 2,
```

```
    0, 0, 0, 0, 0, 2, 2, 0
  ),
  byrow = TRUE, ncol = 8, nrow = 8
)
rownames(A) <- c("a", "b", "s", "c", "d", "e", "f", "z")
colnames(A) <- rownames(A)
wlocal_distances(A, from = "a", to = "d")


A <- matrix(
  c(
    0, 3, 3, 10, 15, 0, 0, 0,
    1, 0, 5, 2, 7, 0, 0, 0,
    3, 5, 0, 0, 0, 0, 0, 0,
    10, 2, 0, 0, 2, 7, 12, 0,
    11, 3, 0, 3, 0, 11, 2, 0,
    0, 0, 0, 7, 11, 0, 3, 2,
    0, 0, 0, 12, 2, 3, 0, 2,
    0, 0, 0, 0, 0, 2, 2, 0
  ),
  byrow = TRUE, ncol = 8, nrow = 8
)
rownames(A) <- c("a", "b", "s", "c", "d", "e", "f", "z")
colnames(A) <- rownames(A)
wall_distances(A, select = "in")
```

---

dist_geographic *Geographical distances*

Description

This function calculate some geographical distances considering a list of places specifying their latitud and longitud. The function currently works for degree decimal or radians formats.

Usage

```
dist_geographic(
  latitude,
  longitud,
  method = c("spherical", "harvesine", "manhattan", "minkowski"),
  places = NULL,
  dd_to_radians = FALSE,
  p = NULL
)
```

Arguments

latitude	A vector with latitude

longitud	A vector with longitud
method	Whether to use the Spherical Law of Cosines spherical (default), Haversine formula harvesine, Manhattan Distance manhattan or Minkowski distance minkowski
places	A vector with the names of the places
dd_to_radians	Whether to transform degree decimal format to radians
p	Parameter p for the estimation of Minkowski distance (default = 2, which is equivalent to an Euclidian Distance)

Value

This function return a distance matrix.

Source

Adapted from Mario Pineda-Krch (Great-circle distance calculations in R)

Examples

```
set.seed(1234)
x <- cbind(latitud = rnorm(5, -90), longitud = rnorm(5, 45))
dist_geographic(x[, 1], x[, 2], method = "harvesine")
```

---

dist_sim_matrix *Structural similarities*

Description

In the literature of social network, Euclidean distance (Burt, 1976) or correlations (Wasserman and Faust, 1994) were considered as measures of structural equivalence.

Usage

```
dist_sim_matrix(
  A,
  method = c("euclidean", "hamming", "jaccard"),
  bipartite = FALSE
)
```

Arguments

A	A matrix
method	The similarities/distance currently available are either Euclidean (default), Hamming, or Jaccard.
bipartite	Whether the object is an incidence matrix

Value

This function returns a distance matrix between nodes of the same matrix.

Author(s)

Alejandro Espinosa-Rada

References

Burt, Ronald S. (1976) Positions in networks. Social Forces, 55(1): 93-122.

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  0, 1, 0, 0, 1,
  0, 0, 0, 1, 1,
  0, 1, 0, 0, 1,
  0, 0, 1, 1, 0,
  0, 1, 0, 0, 0
), nrow = 5, ncol = 5, byrow = TRUE)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- rownames(A)
dist_sim_matrix(A, method = "jaccard")

A <- matrix(c(
  0, 0, 3, 0, 5,
  0, 0, 2, 0, 4,
  5, 4, 0, 4, 0,
  0, 3, 0, 1, 0,
  0, 0, 0, 0, 2
), nrow = 5, ncol = 5, byrow = TRUE)
dist_sim_matrix(A, method = "euclidean")
```

`dyadic_census` *Dyad census*

Description

Dyad census

Usage

```
dyadic_census(G, directed = TRUE, loops = FALSE)
```

Arguments

G	A symmetric matrix object.
directed	Whether the matrix is directed or not
loops	Whether to expect nonzero elements in the diagonal of the matrix

Value

This function return the counts of the dyad census.

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
data(krackhardt_friends)
dyadic_census(krackhardt_friends)

data(FIFAin)
dyadic_census(FIFAin[[1]], directed = FALSE)
```

---

dyad_triad_table *Forbidden triad table*

Description

This function explores dyads and triads (Simmel, 1950), building from the 'forbidden triad' (Granovetter, 1973). First, the minimum structure is an isolated node, then dyads. Afterwards, different combinations of 'forbidden triads' are explored.

Usage

```
dyad_triad_table(A, adjacency_list = FALSE, min = NULL, max = NULL)
```

Arguments

A	A symmetric matrix object.
adjacency_list	Whether to return the adjacency list of triads 201 per node.
min	Numeric constant, lower limit on the size of the triads 201 to find. NULL means no limit, ie. it is the same as 0.
max	Numeric constant, upper limit on the size of the triads 201 to find. NULL means no limit.

Value

This function return the list of triads that each node belong.

If adjacency_list = TRUE it also return the adjacency list of the 'forbidden triads' per node.

Author(s)

Alejandro Espinosa-Rada

References

Granovetter, M.S. (1973). The Strength of Weak Ties. American Journal of Sociology. 78 (6): 1360–80. doi:10.1086/225469.

Simmel, G. (1950). Individual and Society. In K. H. Wolff (Ed.), The Sociology of George Simmel. New York: Free Press.

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  0, 1, 1, 1, 0,
  1, 0, 1, 0, 0,
  1, 1, 0, 0, 0,
  1, 0, 0, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]

dyad_triad_table(A, adjacency_list = TRUE, min = 3)
```

---

eb_constraint *Constraint*

Description

Everett and Borgatti specification of the constraint measure for binary matrices

Usage

```
eb_constraint(A, ego = NULL, digraph = FALSE, weighted = FALSE)
```

Arguments

A	A symmetric matrix object
ego	Name of ego in the matrix
digraph	Whether the matrix is directed or undirected
weighted	Whether the matrix is weighted or not

Value

This function returns term 1, 2 and 3, the normalization and the maximum value of the specification of Everett and Borgatti (2020), and the constraint of Burt (1992).

Author(s)

Alejandro Espinosa-Rada

References

Burt, R.S., 1992. Structural Holes: the Social Structure of Competition. Harvard University Press, Cambridge.

Everett, M.G. and Borgatti, S., 2020. Unpacking Burt's constraint measure. Social Networks 62, pp. 50-57. doi:10.1016/j.socnet.2020.02.001

Examples

```
A <- matrix(c(
  0, 1, 1, 0, 0, 1,
  1, 0, 1, 0, 0, 1,
  1, 1, 0, 0, 0, 1,
  0, 0, 0, 0, 1, 1,
  0, 0, 0, 1, 0, 1,
  1, 1, 1, 1, 1, 0
), ncol = 6, byrow = TRUE)

rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]
eb_constraint(A, ego = "f")
```

---

edgelist_to_matrix *Transform an edgelist to a matrix*

Description

Transform an edgelist to a matrix

Usage

```
edgelist_to_matrix(
  E,
  digraph = TRUE,
  label = NULL,
  label2 = NULL,
  bipartite = FALSE
)
```

Arguments

E	An edge list
digraph	Whether the matrix is directed or not
label	A vector with the names of the nodes
label2	A vector with the names of a different set of nodes
bipartite	Whether the matrix is bipartite

Value

This function transform the edgelist into a matrix

Author(s)

Alejandro Espinosa-Rada

Examples

```
A <- matrix(c(
  0, 1, 1, 0, 0, 0, 0, 1, 0,
  1, 0, 1, 0, 0, 0, 0, 0, 0,
  1, 1, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 1, 1, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 1, 1, 0,
  0, 0, 0, 0, 0, 1, 0, 1, 0,
  1, 0, 0, 0, 0, 1, 1, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0
), byrow = TRUE, ncol = 9)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- rownames(A)
E <- matrix_to_edgelist(A)
edgelist_to_matrix(E, label = c("i"), digraph = FALSE)
```

---

ego_net *Ego network*

Description

Submatrix of ego's neighbourhoods

Usage

```
ego_net(
  A,
  ego = NULL,
  bipartite = FALSE,
  addEgo = FALSE,
  select = c("all", "in", "out")
)
```

Arguments

`A`	A symmetric matrix object
`ego`	Name of ego in the matrix
`bipartite`	Whether the matrix is a two-mode network
`addEgo`	Whether to retain ego in the submatrix or not
`select`	Whether to consider all sender and receiver ties of ego (`all`), only incoming ties (`in`), or outgoing ties (`out`). By default, `all`.

Value

This function returns redundancy, effective size and efficiency measures (Burt, 1992).

Author(s)

Alejandro Espinosa-Rada

References

Burt, R.S., 1992. Structural Holes: the Social Structure of Competition. Harvard University Press, Cambridge.

Borgatti, S., 1997. Unpacking Burt's redundancy measure. Connections, 20(1): 35-38.

Examples

```
A <- matrix(c(
  0, 1, 0, 0, 1, 1, 1,
  1, 0, 0, 1, 0, 0, 1,
  0, 0, 0, 0, 0, 0, 1,
  0, 1, 0, 0, 0, 0, 1,
  1, 0, 0, 0, 0, 0, 1,
  1, 0, 0, 0, 0, 0, 1,
  1, 1, 1, 1, 1, 1, 0
), ncol = 7, byrow = TRUE)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]
ego_net(A, ego = "g")
```

`ei_index` *Krackhardt and Stern's E-I index*

Description

This index was proposed by Krackhardt and Stern (1988) to distinguish between the relative prevalence of between and within-group ties. This measure can be interpreted as homophily at the network level.

Usage

```
ei_index(A, mixed = TRUE, att = NULL)
```

Arguments

A	A symmetric matrix object
mixed	Whether the matrix provided is already a mixed matrix or not
att	Categorical attribute of the nodes

Value

Numerical value of the E-I index.

Examples

```
set.seed(18051889)
n <- 100
A <- matrix(c(rbinom(n, 1, 0.5)),
  ncol = sqrt(n), nrow = sqrt(n), byrow = TRUE
)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]

att <- rbinom(sqrt(n), 3, 0.5)
ei_index(A, mixed = FALSE, att = att)
```

---

expand_matrix *Expand Matrix*

Description

Expand Matrix

Usage

```
expand_matrix(A, label = NULL, loops = FALSE, normalize = FALSE)
```

Arguments

A	A square matrix
label	Duplicated labels to expand the matrix
loops	Whether the loops are retained or not
normalize	Whether to normalize the matrix considering the fractional counting per group

Value

Return an expanded matrix

Author(s)

Alejandro Espinosa-Rada

Examples

```
A <- matrix(c(
  0, 1, 1,
  0, 0, 1,
  1, 0, 0
), byrow = TRUE, ncol = 3, nrow = 3)
rownames(A) <- letters[1:NROW(A)]
colnames(A) <- rownames(A)
label <- sort(rep(rownames(A), 2))
expand_matrix(A, label, loops = FALSE, normalize = TRUE)
```

---

extract_component   *Extract components*

Description

This function extract the matrix of different components

Usage

```
extract_component(A, maximum = TRUE, position = NULL)
```

Arguments

A	A matrix
maximum	Whether to extract the maximum component
position	Whether to extract the component in the ith size position

Value

A matrix or a list of matrices with the required components

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- FIFAex$Matrix
rownames(A) <- FIFAex$label
colnames(A) <- rownames(A)
extract_component(A, maximum = TRUE)
extract_component(A, maximum = FALSE, position = 2)
```

FIFAego *Ego FIFA*

Description

Multilevel Network of the regulatory transnational regime of the International Federation of Association Football (FIFA)

Usage

```
data(FIFAego)
```

Format

A list of a 48 x 48 symmetric matrix of the ego network of FIFA as a different entity, and a string vector with the label of the actors

Source

Espinosa, Alejandro & Ortiz, Francisca (2016). "Jurisdictional autonomy in the regulatory transnational regime of FIFA". REDES- Revista Hispana para el Analisis de Redes Sociales, 27(1), 100-112. (Original title in Spanish: "Autonomia jurisdiccional en el reegimen regulatorio transnacional de la FIFA") doi:10.5565/rev/redes.595

FIFAex *Outside FIFA*

Description

Multilevel Network of the regulatory transnational regime of the International Federation of Association Football (FIFA)

Usage

```
data(FIFAex)
```

Format

A list of a 7 x 7 symmetric matrix of non-FIFA organizations, and a string vector with the label of the actors

Source

Espinosa, Alejandro & Ortiz, Francisca (2016). "Jurisdictional autonomy in the regulatory transnational regime of FIFA". REDES- Revista Hispana para el Analisis de Redes Sociales, 27(1), 100-112. (Original title in Spanish: "Autonomia jurisdiccional en el reegimen regulatorio transnacional de la FIFA") doi:10.5565/rev/redes.595

`FIFAin` *Inside FIFA*

Description

Multilevel Network of the regulatory transnational regime of the International Federation of Association Football (FIFA)

Usage

```
data(FIFAin)
```

Format

A list of a 41 x 41 symmetric matrix of roles and organizations inside FIFA, and a string vector with the label of the actors

Source

Espinosa, Alejandro & Ortiz, Francisca (2016). "Jurisdictional autonomy in the regulatory transnational regime of FIFA". REDES- Revista Hispana para el Analisis de Redes Sociales, 27(1), 100-112. (Original title in Spanish: "Autonomia jurisdiccional en el reegimen regulatorio transnacional de la FIFA") doi:10.5565/rev/redes.595

`fractional_approach` *Fractional approach*

Description

Matrix transformation from incidence matrices to citation networks, fractional counting for co-citation or fractional counting for bibliographic coupling

Usage

```
fractional_approach(
  A1,
  A2,
  approach = c("citation", "cocitation", "bcoupling")
)
```

Arguments

- `A1` — From incidence matrix (e.g. paper and authors)
- `A2` — To incidence matrix (e.g. author to paper)
- `approach` — Character string, “citation”, “cocitation” and “bcoupling”

Value

Return a type of "citation network"

Author(s)

Alejandro Espinosa-Rada

References

Batagelj, V. (2020). Analysis of the Southern women network using fractional approach. Social Networks, 68, 229-236 doi:10.1016/j.socnet.2021.08.001

Batagelj, V., & Cerinšek, M. (2013). On bibliographic networks. Scientometrics, 96(3), 845–864. doi:10.1007/s1119201209401

Examples

```
A1 <- matrix(c(
  1, 0, 0, 0,
  0, 1, 0, 0,
  0, 1, 1, 1,
  0, 0, 0, 0,
  0, 0, 0, 1
), byrow = TRUE, ncol = 4)

A2 <- matrix(c(
  1, 1, 1, 0, 0,
  0, 0, 1, 0, 0,
  0, 0, 1, 1, 0,
  0, 0, 0, 1, 1
), byrow = TRUE, ncol = 5)

fractional_approach(A1, A2)
```

`gen_degree` *Generalized degree*

Description

Generalized degree centrality for one-mode and bipartite networks

Usage

```
gen_degree(
  A,
  weighted = FALSE,
  type = "out",
  normalized = FALSE,
  loops = TRUE,
  digraph = TRUE,
  alpha = 0.5,
  bipartite = FALSE
)
```

Arguments

`A`	A matrix object
`weighted`	Whether the matrix is weighted or not
`type`	Character string, “out” (outdegree), “in” (indegree) and “all” (degree)
`normalized`	Whether normalize the measure for the one-mode network (Freeman, 1978) or a bipartite network (Borgatti and Everett, 1997)
`loops`	Whether the diagonal of the matrix is considered or not
`digraph`	Whether the matrix is directed or undirected
`alpha`	Sets the alpha parameter in the generalised measures from Opsahl et al. (2010)
`bipartite`	Whether the matrix is bipartite or not.

Value

This function returns term 1, 2 and 3, the normalization and the maximum value of the specification of Everett and Borgatti (2020), and the constraint of Burt (1992)

Author(s)

Alejandro Espinosa-Rada

References

Borgatti, S. P., and Everett, M. G. (1997). Network analysis of 2-mode data. Social Networks, 19(3), 243–269.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239.

Opsahl, T., Agneessens, F., and Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), 245–251.

Examples

```
A3 <- matrix(c(
  0, 4, 4, 0, 0, 0,
  4, 0, 2, 1, 1, 0,
  4, 2, 0, 0, 0, 0,
  0, 1, 0, 0, 0, 0,
  0, 1, 0, 0, 0, 7,
  0, 0, 0, 0, 7, 0
), byrow = TRUE, ncol = 6)

gen_degree(A3, digraph = FALSE, weighted = TRUE)
```

---

gen_density *Generalized density*

Description

Generalized density

Usage

```
gen_density(
  A,
  directed = TRUE,
  bipartite = FALSE,
  loops = FALSE,
  weighted = FALSE,
  multilayer = FALSE
)
```

Arguments

A	A symmetric or incidence matrix object
directed	Whether the matrix is directed
bipartite	Whether the matrix is bipartite
loops	Whether to consider the loops
weighted	Whether the matrix is weighted
multilayer	Whether the matrix is multilayer (i.e., multiplex and/or multilevel)

Value

This function returns the density of the matrix(es)

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S., and Faust, K. (1994). Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press.

Examples

```
# A bipartite matrix
B <- matrix(c(
  1, 1, 0,
  0, 0, 1,
  0, 1, 1,
  0, 0, 1
), byrow = TRUE, ncol = 3)
gen_density(B, bipartite = TRUE)

# A multilevel network
A1 <- matrix(c(
  0, 1, 0, 0, 1,
  1, 0, 0, 1, 1,
  0, 0, 0, 1, 1,
  0, 1, 1, 0, 1,
  1, 1, 1, 1, 0
), byrow = TRUE, ncol = 5)

B1 <- matrix(c(
  1, 0, 0,
  1, 1, 0,
  0, 1, 0,
  0, 1, 0,
  0, 1, 1
), byrow = TRUE, ncol = 3)

A2 <- matrix(c(
  0, 1, 1,
  1, 0, 0,
  1, 0, 0
), byrow = TRUE, nrow = 3)

B2 <- matrix(c(
  1, 1, 0, 0,
  0, 0, 1, 0,
  0, 0, 1, 1
), byrow = TRUE, ncol = 4)

A3 <- matrix(c(
  0, 1, 3, 1,
  1, 0, 0, 0,
  3, 0, 0, 5,
  1, 0, 5, 0
), byrow = TRUE, ncol = 4)

matrices <- list(A1, B1, A2, B2, A3)
```

```
gen_density(matrices, multilayer = TRUE)

# A multiplex network
A <- matrix(c(
  0, 1, 3, 6, 4,
  2, 0, 4, 5, 2,
  4, 1, 0, 6, 1,
  5, 6, 3, 0, 6,
  1, 1, 2, 3, 0
), byrow = TRUE, ncol = 5)
gen_density(A, multilayer = TRUE)
```

`heterogeneity` *Blau's and IQV index*

Description

This index was used by Blau (1977) to distinguish between the relative prevalence of between and within-group ties. This measure can be interpreted as heterogeneity at the network level.

Usage

```
heterogeneity(att, normalized = FALSE)
```

Arguments

`att`	Categorical attribute of the nodes
`normalized`	Whether to return IQV index

Value

Numerical value of the Blau index.

If `normalized = TRUE`, then the function also return IQV index.

References

Agresti, A. and Agresti, B. (1978). Statistical Analysis of Qualitative Variation. Sociological Methodology, 9, 204-237. doi:10.2307/270810

Blau, P. M. (1977). Inequality and heterogeneity. New York: Free Press.

Examples

```
a <- rep(1:10, 10)
heterogeneity(a, normalized = TRUE)

a <- rep(1:2, 10)
heterogeneity(a, normalized = TRUE)
```

`hypergraph` *Hypergraphs*

Description

Hypergraph consist of a set of objects and a collection of subsets of objects, in which each object belongs to at least one subset, and no subset is empty (Berge, 1989)

Usage

```
hypergraph(A, dual = TRUE, both = TRUE)
```

Arguments

`A`	An incidence matrix.
`dual`	Whether to return the dual hypergraph (which rever the role of the pointes and the edges)
`both`	Whether to return the hypergraph and the dual hypergraph

Value

This function returns an adjacent list of the subsets of entities in the hypergraph.

Author(s)

Alejandro Espinosa-Rada

References

Berge, C. (1973). Graphs and hypergraphs.Amsterdam: North-Holland.

Berge, C. (1989). Hypergraphs: Combinatorics of finite sets. Amsterdam: North-Holland.

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  1, 0, 1,
  0, 1, 0,
  0, 1, 1,
  0, 0, 1,
  1, 1, 1,
  1, 1, 0
), byrow = TRUE, ncol = 3)
colnames(A) <- letters[1:ncol(A)]
rownames(A) <- letters[(ncol(A) + 1):(nrow(A) + ncol(A))]
hypergraph(A, both = TRUE)
```

ind_rand_matrix *Independent random matrix*

Description

The function creates random matrices following a uniform probability over the space of networks having exactly m fixed number of edges (Moreno and Jennings, 1938; Rapoport, 1948; Solomonoff and Rapoport, 1951; Erdos and Renyi, 1959) or following a probability of the formation of the ties (Gilbert, 1959) assuming ties independency.

Usage

```
ind_rand_matrix(
  n,
  m = NULL,
  type = c("edges", "probability"),
  digraph = TRUE,
  loops = FALSE,
  l = NULL,
  p = NULL,
  trials = 1,
  multilevel = FALSE
)
```

Arguments

`n`	The number of nodes of the first set
`m`	The number of nodes of a second set
`type`	The model assumes a fixed number of `edges` model (a.k.a. G(n,m)) (default) or a `probability` model (a.k.a. G(n,p))
`digraph`	Whether the matrix is symmetric or not
`loops`	Whether to expect nonzero elements in the diagonal of the matrix
`l`	The number of ties expected for the `edges` (a.k.a. G(n,m)) model
`p`	The probability of the ties expected for the `probability` (a.k.a. G(n,p)) model. If no parameter ‘p‘ is specified, a uniform distribution is considered (p=0.5).
`trials`	Whether to add counting numbers to the `probability` (a.k.a. G(n,p)) model
`multilevel`	Whether to return a meta-matrix to represent a multilevel network

Details

The fixed model is often called the G(n,m) graph with ’n’ nodes and ’m’ edges, and the ’m’ edges are chosen uniformly randomly from the set of all possible ties.

The probability model is known as the G(n,p) graph, in which the matrix has ’n’ nodes, and for each tie, the probability that it is present in the matrix is ’p’.

These are the simplest models that follow a conditional uniform distribution that place nonnull probability on a subset of networks with distinctive characteristics corresponding to the observed networks - for example, simulating a matrix based on the number of ties observed in the network.

Value

This function return the counts of the dyad census.

Author(s)

Alejandro Espinosa-Rada

References

Erdos, P. and Renyi, A. (1959). On random graphs. Publicationes Mathematicae 6, 290–297.

Gilbert, N. (1959). Random Graphs. The Annals of Mathematical Statistics, 30(4): 1141-1144.

Moreno, J. and Jennings, H. (1938). Statistics of social configurations. Sociometry, 1(3/4):342–374.

Rapoport, A. (1948). Cycle distributions in random nets. Bulletin of Mathematical Biology, 10(3):145–157.

Solomonoff, R. and Rapoport, A. (1951). Connectivity of random nets. Bulletin of Mathematical Biology, 13:107–117.

Examples

```
set.seed(18051889)
ind_rand_matrix(5, type = "edges", l = 3, digraph = TRUE, loops = TRUE)
ind_rand_matrix(5, type = "probability")
ind_rand_matrix(n = 5, m = 2, p = 0.20, type = "probability", multilevel = TRUE)
```

---

jaccard *Jaccard similarity*

Description

Jaccard similarity identifies the changes of ties between two matrices.

Usage

```
jaccard(
  A,
  B,
  directed = TRUE,
  diag = FALSE,
  coparticipation = FALSE,
  bipartite = FALSE
)
```

Arguments

A	Binary matrix A
B	Binary matrix B
directed	Whether the matrix is directed (asymmetric)
diag	Whether the diagonal should be considered
coparticipation	Select nodes that co-participate in both matrices
bipartite	Whether the matrix is incidence

Value

The output are: `jaccard` = Jaccard similarity, `proportion` = proportion among the ties present at a given observation of ties that are also present in the other matrix, and `table` = a table with the tie changes between matrices.

If `coparticipation` = `TRUE`, then also: `match` = The number of nodes present in both matrices; `size_matrix1` = The size of the first matrix; `size_matrix2` = The size of the second matrix; `coparticipation1` = The percentage of nodes in the first matrix also present in the second matrix; `coparticipation2` = The percentage of nodes in the second matrix also present in the first matrix: `overlap_actors` = Overlap of nodes between two matrices

If `coparticipation` = `TRUE` and `bipartite` = `TRUE`, then also: `matchM1` = The number of nodes in the first 'mode' present in both matrices; `matchM2` = The number of nodes in the second 'mode' present in both matrices; `size_matrix1_M1` = The number of nodes in the first 'mode' of the first matrix; `size_matrix1_M2` = The number of nodes in the second 'mode' of the first matrix; `size_matrix2_M1` = The number of nodes in the first 'mode' of the second matrix; `size_matrix2_M2` = The number of nodes in the second 'mode' of the second matrix; `coparticipation1_M2` = The percentage of nodes of the first 'mode' in the first matrix present in the second matrix. `coparticipation1_M2` = The percentage of nodes of the second 'mode' in the first matrix present in the second matrix. `coparticipation2_M1` = The percentage of nodes of the first 'mode' in the second matrix present in the first matrix. `coparticipation2_M2` = The percentage of nodes of the second 'mode' in the second matrix present in the first matrix. `overlap_actors_M1` = Overlap between two matrices (nodes of the first 'mode') `overlap_actors_M2` = Overlap between two matrices (nodes of the second 'mode')

Author(s)

Alejandro Espinosa-Rada

References

Batagelj, V., and Bren, M. (1995). Comparing resemblance measures. Journal of Classification 12, 73–90.

Examples

```
A <- matrix(c(
  0, 1, 1, 0,
  1, 0, 0, 0,
```

```
  1, 0, 0, 0,
  0, 0, 1, 0
), byrow = TRUE, ncol = 4)
B <- matrix(c(
  0, 1, 1, 0,
  1, 0, 0, 0,
  1, 0, 0, 0,
  0, 0, 0, 0
), byrow = TRUE, ncol = 4)
jaccard(A, B, directed = TRUE)
```

`kp_reciprocity` *Reciprocity of Katz and Powell*

Description

Reciprocity of Katz and Powell

Usage

```
kp_reciprocity(G, fixed = FALSE, d = NULL, dichotomic = TRUE)
```

Arguments

`G`	A symmetric matrix object.
`fixed`	Whether the choices are fixed or not
`d`	Numeric value of the number of fixed choices.
`dichotomic`	Whether the matrix is weighted or binary

Value

This function gives a measurement of the tendency toward reciprocation of choices.

Author(s)

Alejandro Espinosa-Rada

References

Katz, L. and Powell, J.H. (1955). "Measurement of the tendency toward reciprocation of choice." Sociometry, 18:659-665.

Examples

```
data(krackhardt_friends)
kp_reciprocity(krackhardt_friends, fixed = TRUE, d = 5)
```

krackhardt_friends *Krackhardt friends*

Description

Friendship network of the relations measured for Krackhardt's high-tech managers.

Usage

```
data(krackhardt_friends)
```

Format

A 21 x 21 directed matrix of the managers

Source

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

k_core *Generalized k-core*

Description

Generalized k-core for undirected, directed, weighted and multilevel networks

Usage

```
k_core(
  A,
  B1 = NULL,
  multilevel = FALSE,
  type = "in",
  digraph = FALSE,
  loops = FALSE,
  weighted = FALSE,
  alpha = 1
)
```

Arguments

A	A matrix object.
B1	An incidence matrix for multilevel networks.
multilevel	Whether the measure of k-core is for multilevel networks.
type	Character string, “out” (outdegree), “in” (indegree) and “all” (degree)
digraph	Whether the matrix is directed or undirected
loops	Whether the diagonal of the matrix is considered or not
weighted	Whether the measure of k-core is for valued matrices
alpha	Sets the alpha parameter in the generalised measures from Opsahl et al. (2010)

Value

This function return the k-core.

Author(s)

Alejandro Espinosa-Rada

References

Batagelj, V., & Zaveršnik, M. (2011). Fast algorithms for determining (generalized) core groups in social networks. Advances in Data Analysis and Classification, 5(2), 129–145. doi:10.1007/s116340100079y

Eidsaa, M., & Almaas, E. (2013). s-core network decomposition: A generalization of k-core analysis to weighted networks. Physical Review E, 88(6), 062819. doi:10.1103/PhysRevE.88.062819

Seidman S (1983). 'Network structure and minimum degree'. Social Networks, 5, 269-287.

Examples

```
A1 <- matrix(c(
  0, 1, 0, 0, 0,
  1, 0, 0, 1, 0,
  0, 0, 0, 1, 0,
  0, 1, 1, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)
B1 <- matrix(c(
  1, 0, 0,
  1, 1, 0,
  0, 1, 0,
  0, 1, 0,
  0, 1, 1
), byrow = TRUE, ncol = 3)

k_core(A1, B1, multilevel = TRUE)
```

`lazega_lawfirm` *Lazega law firm*

Description

The data is part of a study carried out in a Northeastern US corporate law firm, referred to as SG&R, 1988-1991 in New England. The data were collected by Emmanuel Lazega (2001). This is a multiplex network of attorneys (partners and associates) of this firm. It includes (among others) measurements of networks among the 71 attorneys (partners and associates) of this firm, i.e. their strong-coworker network, advice network, friendship network, and indirect control networks.

Usage

```
data(lazega_lawfirm)
```

Format

Three 71 X 71 matrices:

cowork A matrix indicating "cowork" relationships among the attorneys, based on their work together on cases and other professional activities.

advice A matrix indicating "advice" relationships, where the matrix shows to whom attorneys went for professional advice.

friends A matrix indicating "friends" relationships, showing social connections outside of work.

attributes A data frame with attributes of the actors, including:

- **seniority** Seniority.
- **status** 1=partner; 2=associate
- **gender** 1=man; 2=woman
- **office** 1=Boston; 2=Hartford; 3=Providence
- **years** Years with the firm
- **age** Age
- **practice** 1=litigation; 2=corporate
- **law_school** 1 = Harvard, Yale; 2 = UConn; 3 = Other

Source

Lazega, Emmanuel (2001) The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership. Oxford University Press.

`matrix_adjlist` *Transform a matrix to an adjacency list*

Description

Transform a matrix to an adjacency list

Usage

```
matrix_adjlist(A)
```

Arguments

A	A matrix

Value

This function transform a matrix to an adjacency list

Author(s)

Alejandro Espinosa-Rada

Examples

```
A <- matrix(c(
  0, 1, 1, 0, 0, 0, 0, 1, 0,
  1, 0, 1, 0, 0, 0, 0, 0, 0,
  1, 1, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 1, 1, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 1, 1, 0,
  0, 0, 0, 0, 0, 1, 0, 1, 0,
  1, 0, 0, 0, 0, 1, 1, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0
), byrow = TRUE, ncol = 9)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- rownames(A)
matrix_adjlist(A)
```

matrix_projection *Unipartite projections*

Description

Two-mode networks can be represented (or 'projected') as one-mode networks.

Usage

```
matrix_projection(A, B = NULL, digraph = FALSE)
```

Arguments

A	A first matrix object
B	A second matrix object
digraph	Whether the matrix is directed or not

Value

This function return a list of matrices of the two projections of the original matrix.

Author(s)

Alejandro Espinosa-Rada

References

Davis, Allison; Gardner, Burleigh B. and Mary. R. Gardner (1941). Deep South: A Social Anthropological Study of Caste and Class. The University of Chicago Press, Chicago.

Breiger, Ronald L. (1976). The Duality of Persons and Groups, 53(2), 181-190 doi:10.2307/2576011

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  2, 0, 2,
  1, 1, 0,
  0, 3, 3,
  0, 2, 2,
  0, 0, 1
), byrow = TRUE, ncol = 3)
matrix_projection(A)

A <- matrix(c(
  0, 0, 0, 0, 1,
```

```
  1, 0, 0, 0, 0,
  1, 1, 0, 0, 0,
  0, 1, 1, 1, 1,
  0, 0, 1, 0, 0,
  0, 0, 1, 1, 0
), byrow = TRUE, ncol = 5)

B <- matrix(c(
  0, 0, 0, 0, 1,
  1, 0, 0, 0, 0,
  1, 0, 0, 0, 0,
  0, 1, 0, 0, 0,
  0, 0, 1, 0, 0,
  0, 0, 1, 0, 0
), byrow = TRUE, ncol = 5)
matrix_projection(A, B, digraph = TRUE)
```

`matrix_report` *Matrix report*

Description

The primary matrix used in social network analysis are the adjacency matrix or sociomatrix, and the incidence matrix.

Usage

```
matrix_report(A)
```

Arguments

A	A matrix

Value

This function return a report of some of the characteristics of the matrix.

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  1, 1, 0, 0, -1,
  1, 0, 0, 1, 1,
  0, 0, NA, 1, 1,
  0, 1, 1, 0, 1,
  1, 1, 1, 1, 0
), byrow = TRUE, ncol = 5)

B <- matrix(c(
  1, 0, 0,
  1, 1, 0,
  0, NA, 0,
  0, 1, 0,
  0, 1, 1
), byrow = TRUE, ncol = 3)
matrix_report(A)
matrix_report(B)
```

---

matrix_to_edgelist *Transform a square matrix to an edge-list*

Description

Transform a square matrix to an edge-list

Usage

```
matrix_to_edgelist(A, digraph = FALSE, valued = FALSE, loops = FALSE)
```

Arguments

A	A square matrix
digraph	Whether the matrix is directed or not
valued	Add a third columns with the valued of the relationship
loops	Whether the loops are retained or not

Value

This function transform the matrix into an edgelist

Author(s)

Alejandro Espinosa-Rada

Examples

```
A <- matrix(c(
  0, 2, 1,
  1, 0, 0,
  1, 0, 1
), byrow = TRUE, ncol = 3)
matrix_to_edgelist(A, digraph = TRUE, valued = TRUE, loops = TRUE)
```

---

meta_matrix — *Meta matrix for multilevel networks*

Description

Meta matrix for multilevel networks

Usage

```
meta_matrix(A1, B1, A2 = NULL, B2 = NULL, A3 = NULL, B3 = NULL)
```

Arguments

A1	The square matrix of the lowest level
B1	The incidence matrix of the ties between the nodes of first level and the nodes of the second level
A2	The square matrix of the second level
B2	The incidence matrix of the ties between the nodes of the second level and the nodes of the third level
A3	The square matrix of the third level
B3	The incidence matrix of the ties between the nodes of the third level and the nodes of the first level

Value

Return a meta matrix for multilevel networks

Author(s)

Alejandro Espinosa-Rada

References

Carley, K. M. (2002). Smart agents and organizations of the future. In: Leah Lievrouw & Sonia Livingstone (Eds.), The Handbook of New Media (pp. 206-220). Thousand Oaks, CA, Sage.

Krackhardt, D., & Carley, K. M. (1998). PCANS model of structure in organizations (pp. 113-119). Pittsburgh, Pa, USA: Carnegie Mellon University, Institute for Complex Engineered Systems.

Examples

```
A1 <- matrix(c(
  0, 1, 0, 0, 0,
  1, 0, 0, 1, 0,
  0, 0, 0, 1, 0,
  0, 1, 1, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)

B1 <- matrix(c(
  1, 0, 0,
  1, 1, 0,
  0, 1, 0,
  0, 1, 0,
  0, 1, 1
), byrow = TRUE, ncol = 3)

A2 <- matrix(c(
  0, 1, 1,
  1, 0, 0,
  1, 0, 0
), byrow = TRUE, nrow = 3)

B2 <- matrix(c(
  1, 1, 0, 0,
  0, 0, 1, 0,
  0, 0, 1, 1
), byrow = TRUE, ncol = 4)

A3 <- matrix(c(
  0, 1, 1, 1,
  1, 0, 0, 0,
  1, 0, 0, 1,
  1, 0, 1, 0
), byrow = TRUE, ncol = 4)

B3 <- matrix(c(
  1, 0, 0, 0, 0,
  0, 1, 0, 1, 0,
  0, 0, 0, 0, 0,
  0, 0, 0, 0, 0
), byrow = TRUE, ncol = 5)

rownames(A1) <- letters[1:nrow(A1)]
colnames(A1) <- rownames(A1)
rownames(A2) <- letters[nrow(A1) + 1:nrow(A2)]
colnames(A2) <- rownames(A2)
rownames(B1) <- rownames(A1)
colnames(B1) <- colnames(A2)
rownames(A3) <- letters[nrow(A1) + nrow(A2) + 1:nrow(A3)]
colnames(A3) <- rownames(A3)
rownames(B2) <- rownames(A2)
```

```
colnames(B2) <- colnames(A3)
rownames(B3) <- rownames(A3)
colnames(B3) <- rownames(A1)
meta_matrix(A1, B1, A2, B2, A3, B3)
```

`minmax_overlap` *Minimum/maximum overlap*

Description

Two-mode networks can be represented (or 'projected') as one-mode networks.

Usage

```
minmax_overlap(A, row = TRUE, min = TRUE)
```

Arguments

`A`	A matrix object
`row`	Whether to consider the actors in the rows of the matrix (default) or the column.
`min`	Whether to extract the minimum (default) or the maximum overlap.

Value

This function return the overlap between the modes (a.k.a. actors, nodes, vertices).

Author(s)

Alejandro Espinosa-Rada

References

Morris, S.A. (2005). Unified Mathematical Treatment of Complex Cascaded Bipartite Networks: The Case of Collections of Journal Papers. Unpublished PhD Thesis, Oklahoma State University.

Examples

```
A <- matrix(c(
  2, 0, 2,
  1, 1, 0,
  0, 3, 3,
  0, 2, 2,
  0, 0, 1
), byrow = TRUE, ncol = 3)
minmax_overlap(A)
```

mixed_census *Multilevel triad and quadrilateral census*

Description

Multilevel triad and quadrilateral census

Usage

```
mixed_census(A1, B1, B2 = NULL, quad = FALSE)
```

Arguments

A1	An adjacent matrix object.
B1	An incidence matrix object.
B2	An incidence matrix object.
quad	Whether the matrix is a quadrilateral census or not.

Value

This function return the counts of a multilevel census.

If quad = TRUE, then the function return the multilevel quadrilateral census.

Author(s)

Alejandro Espinosa-Rada

References

Espinosa-Rada, A. (2021). A Network Approach for the Sociological Study of Science: Modelling Dynamic Multilevel Networks. [PhD](https://research.manchester.ac.uk/en/studentTheses/a-network-approach-for-the-sociological-study-of-science-and-know). The University of Manchester.

Espinosa-Rada, A., Bellotti, E., Everett, M., & Stadtfeld, C. (2024). Co-evolution of a socio-cognitive scientific network: A case study of citation dynamics among astronomers. Social Networks, 78, 92–108. doi:10.1016/j.socnet.2023.11.008

Hollway, J., Lomi, A., Pallotti, F., & Stadtfeld, C. (2017). Multilevel social spaces: The network dynamics of organizational fields. Network Science, 5(2), 187–212. doi:10.1017/nws.2017.8

Examples

```
B1 <- matrix(c(
  1, 1, 0,
  0, 0, 1,
  0, 0, 1,
  1, 0, 0
), byrow = TRUE, ncol = 3)
```

```
A1 <- matrix(c(
  0, 1, 0, 1,
  1, 0, 0, 1,
  0, 1, 0, 1,
  1, 0, 1, 0
), byrow = TRUE, ncol = 4)
B2 <- matrix(c(
  1, 0, 0, 0, 0,
  0, 1, 0, 1, 0,
  0, 0, 0, 0, 0,
  0, 0, 0, 0, 0
), byrow = TRUE, ncol = 5)

mixed_census(A1, B1, B2, quad = TRUE)
```

---

mix_matrix *Mixing matrix*

Description

Create a mixing matrix from node attributes. The mixing matrix is a two-dimensional matrix that cross-classifies the edges depending on the values of their attributes. This matrix allowed identifying segregation and homophily at the network level.

Usage

```
mix_matrix(A, att = NULL)
```

Arguments

A	A symmetric matrix object
att	Categorical attribute of the nodes

Details

Values in the diagonal are the number of ties within groups, and off-diagonal are the number of relations between groups.

Value

This function returns a mixing matrix.

Author(s)

Alejandro Espinosa-Rada

Examples

```
n <- 100
A <- matrix(c(rbinom(n, 1, 0.5)),
  ncol = sqrt(n), nrow = sqrt(n), byrow = TRUE
)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]
att <- rbinom(sqrt(n), 3, 0.5)
mix_matrix(A, att = att)
```

---

multilevel_degree *Degree centrality for multilevel networks*

Description

Degree centrality for multilevel networks

Usage

```
multilevel_degree(
  A1,
  B1,
  A2 = NULL,
  B2 = NULL,
  A3 = NULL,
  B3 = NULL,
  complete = FALSE,
  digraphA1 = FALSE,
  digraphA2 = FALSE,
  digraphA3 = FALSE,
  typeA1 = "out",
  typeA2 = "out",
  typeA3 = "out",
  loopsA1 = FALSE,
  loopsA2 = FALSE,
  loopsA3 = FALSE,
  normalized = FALSE,
  weightedA1 = FALSE,
  weightedA2 = FALSE,
  weightedA3 = FALSE,
  alphaA1 = 0.5,
  alphaA2 = 0.5,
  alphaA3 = 0.5
)
```

Arguments

A1	The square matrix of the lowest level
B1	The incidence matrix of the ties between the nodes of first level and the nodes of the second level
A2	The square matrix of the second level
B2	The incidence matrix of the ties between the nodes of the second level and the nodes of the third level
A3	The square matrix of the third level
B3	The incidence matrix of the ties between the nodes of the third level and the nodes of the first level
complete	Add the degree of bipartite and tripartite networks for B1, B2 and/or B3, and the low_multilevel (i.e. A1+B1+B2+B3), meso_multilevel (i.e. B1+A2+B2+B3) and high_multilevel (i.e. B1+B2+A3+B3) degree
digraphA1	Whether A1 is a directed network
digraphA2	Whether A2 is a directed network
digraphA3	Whether A3 is a directed network
typeA1	Type of degree of the network for A1, "out" for out-degree, "in" for in-degree or "all" for the sum of the two
typeA2	Type of degree of the network for A2, "out" for out-degree, "in" for in-degree or "all" for the sum of the two
typeA3	Type of degree of the network for A3, "out" for out-degree, "in" for in-degree or "all" for the sum of the two
loopsA1	Whether the loops of the edges are considered in matrix A1
loopsA2	Whether the loops of the edges are considered in matrix A2
loopsA3	Whether the loops of the edges are considered in matrix A3
normalized	If TRUE then the result is divided by (n-1)+k+m for the first level, (m-1)+n+k for the second level, and (k-1)+m+n according to Espinosa-Rada et al. (2021)
weightedA1	Whether A1 is weighted
weightedA2	Whether A2 is weighted
weightedA3	Whether A3 is weighted
alphaA1	The alpha parameter of A1 according to Opsahl et al (2010) for weighted networks. The value 0.5 is given by default.
alphaA2	The alpha parameter of A2 according to Opsahl et al (2010) for weighted networks. The value 0.5 is given by default.
alphaA3	The alpha parameter of A3 according to Opsahl et al (2010) for weighted networks. The value 0.5 is given by default.

Value

Return a data.frame of multilevel degree

Author(s)

Alejandro Espinosa-Rada

References

Borgatti, S. P., and Everett, M. G. (1997). Network analysis of 2-mode data. Social Networks, 19(3), 243–269.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239.

Opsahl, T., Agneessens, F., and Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), 245–251.

Examples

```
A1 <- matrix(c(
  0, 1, 0, 0, 0,
  1, 0, 0, 1, 0,
  0, 0, 0, 1, 0,
  0, 1, 1, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)

B1 <- matrix(c(
  1, 0, 0,
  1, 1, 0,
  0, 1, 0,
  0, 1, 0,
  0, 1, 1
), byrow = TRUE, ncol = 3)

A2 <- matrix(c(
  0, 1, 1,
  1, 0, 0,
  1, 0, 0
), byrow = TRUE, nrow = 3)

B2 <- matrix(c(
  1, 1, 0, 0,
  0, 0, 1, 0,
  0, 0, 1, 1
), byrow = TRUE, ncol = 4)

A3 <- matrix(c(
  0, 1, 1, 1,
  1, 0, 0, 0,
  1, 0, 0, 1,
  1, 0, 1, 0
), byrow = TRUE, ncol = 4)

B3 <- matrix(c(
  1, 0, 0, 0, 0,
```

```
  0, 1, 0, 1, 0,
  0, 0, 0, 0, 0,
  0, 0, 0, 0, 0
), byrow = TRUE, ncol = 5)

multilevel_degree(A1, B1, A2, B2, A3, B3)

multilevel_degree(A1, B1, A2, B2, A3, B3, normalized = TRUE, complete = TRUE)
```

---

multiplex_census *Multiplex triad census*

Description

This function counts the different subgraphs of three nodes in a multiplex directed and undirected network.

Usage

```
multiplex_census(A, B)
```

Arguments

A	A directed matrix object.
B	An undirected matrix object.

Value

This function gives the counts of the mixed multiplex triad census for a directed and an undirected network.

Author(s)

Alejandro Espinosa-Rada

References

Espinosa-Rada, A. (2021). A Network Approach for the Sociological Study of Science: Modelling Dynamic Multilevel Networks. [PhD](https://research.manchester.ac.uk/en/studentTheses/a-network-approach-for-the-sociological-study-of-science-and-know). The University of Manchester.

Espinosa-Rada, A., Bellotti, E., Everett, M., & Stadtfeld, C. (2024). Co-evolution of a socio-cognitive scientific network: A case study of citation dynamics among astronomers. Social Networks, 78, 92–108. doi:10.1016/j.socnet.2023.11.008

Examples

```
# SOAR
A <- matrix(
  c(
    0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1,
    0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
    0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1,
    0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  ),
  byrow = TRUE, ncol = 12
)

B <- matrix(
  c(
    0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
    0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  ),
  byrow = TRUE, ncol = 12
)

multiplex_census(A, B)
```

---

percolation_clique *Clique percolation*

Description

Clique Percolation Method (CPM) is an algorithm for finding overlapping communities within networks, introduced by Palla et al. (2005). This function firstly identify cliques of size k, then creates a incidence matrix as an affiliation network.

Usage

```
percolation_clique(A)
```

Arguments

A	A matrix

Value

A matrix that assign each node to a clique

Author(s)

Alejandro Espinosa-Rada

References

Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435(7043), 814-818.

Examples

```
A <- matrix(
  c(
    0, 1, 1, 1, 0, 0, 0, 0, 0,
    1, 0, 1, 0, 0, 0, 0, 0, 0,
    1, 1, 0, 1, 0, 0, 0, 0, 0,
    1, 0, 1, 0, 1, 1, 0, 0, 0,
    0, 0, 0, 1, 0, 1, 1, 1, 0,
    0, 0, 0, 1, 1, 0, 1, 1, 0,
    0, 0, 0, 0, 1, 1, 0, 1, 1,
    0, 0, 0, 0, 1, 1, 1, 0, 0,
    0, 0, 0, 0, 0, 0, 1, 0, 0
  ),
  byrow = TRUE, ncol = 9
)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]
percolation_clique(A)
```

---

perm_label   *Permute labels of a matrix*

Description

This function permutes the labels of a matrix.

Usage

```
perm_label(A, m = 1, unique = FALSE)
```

Arguments

A	A matrix
m	Number of permutations
unique	Whether to return unique cases

Value

This function returns the permutation of labels.

Author(s)

Alejandro Espinosa-Rada

Examples

```
W <- matrix(c(
  0, 1, 0, 0, 0,
  0, 0, 1, 0, 0,
  1, 0, 0, 0, 0,
  0, 0, 0, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)
rownames(W) <- c("P", "Q", "R", "S", "T")
colnames(W) <- rownames(W)
perm_label(W, m = 1000, unique = TRUE)
```

---

perm_matrix *Permutation matrix*

Description

This function create permutation matrices.

Usage

```
perm_matrix(n, m = 1, unique = FALSE)
```

Arguments

n	The size of the square matrix
m	Number of permutations
unique	Whether to return unique cases

Value

This function returns a list of permutation matrices

Author(s)

Alejandro Espinosa-Rada

Examples

```
W <- matrix(c(
  0, 1, 0, 0, 0,
  0, 0, 1, 0, 0,
  1, 0, 0, 0, 0,
  0, 0, 0, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)
rownames(W) <- c("P", "Q", "R", "S", "T")
colnames(W) <- rownames(W)
perm_matrix(5, m = 1000, unique = TRUE)
```

---

posneg_index *Positive-negative centrality*

Description

Positive-negative centrality

Usage

```
posneg_index(A, select = c("all", "in", "out"))
```

Arguments

A	A signed symmetric matrix (i.e., with ties that are either -1, 0 or 1)
select	Whether to consider the direction of the outgoing ties. Considering all (default), in or out ties.

Value

This function return the positive-negative centrality index for signed networks (Everett and Borgatti).

Source

Adapted from David Schoch 'signnet'

References

Everett, Martin and Borgatti, Stephen (2014). Networks containing negative ties. Social Networks, 38, 111-120. doi:10.1016/j.socnet.2014.03.005

Examples

```
A <- matrix(
  c(
    0, 1, -1, -1, -1, -1, 0, 0, 0, 0, 0, -1, 0, 0, 1, 1,
    1, 0, -1, 0, -1, -1, 0, 0, -1, -1, 0, 0, 0, 0, 1, 1,
    -1, -1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
    -1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
    -1, -1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, -1, -1,
    -1, -1, 1, 0, 0, 0, 1, 1, -1, 0, 1, 1, -1, 0, 0, -1,
    0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0,
    0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, -1, 0, 0,
    0, -1, 0, 0, 1, -1, 0, 0, 0, 1, -1, 0, 1, 0, -1, 0,
    0, -1, 0, 0, 0, 0, 0, 0, 1, 0, -1, 0, 1, 0, -1, 0,
    0, 0, 0, 0, 0, 1, 1, 1, -1, -1, 0, 1, -1, 0, -1, -1,
    -1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, -1, -1, -1,
    0, 0, 0, 0, 0, -1, 1, 0, 1, 1, -1, 0, 0, 1, -1, -1,
    0, 0, 0, 0, 1, 0, 0, -1, 0, 0, 0, -1, 1, 0, 0, -1,
    1, 1, 0, 0, -1, 0, 0, 0, -1, -1, -1, -1, -1, 0, 0, 1,
    1, 1, 0, 0, -1, -1, 0, 0, 0, 0, -1, -1, -1, -1, 1, 0
  ),
  ncol = 16, nrow = 16, byrow = TRUE
)
label <- c(
  "Gavev", "Kotun", "Ove", "Alika", "Nagam", "Gahuk", "Masil", "Ukudz",
  "Notoh", "Kohik", "Geham", "Asaro", "Uheto", "Seuve", "Nagad", "Gama"
)
rownames(A) <- label
colnames(A) <- rownames(A)
posneg_index(A, select = c("all"))
```

---

power_function *Power matrix*

Description

Power of a matrix computed by successive matrix multiplication.

Usage

```
power_function(A, n)
```

Arguments

A	A matrix
n	Positive integer

Value

This function return the power of a matrix by repeating matrix multiplication.

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  1, 0, 0, 0,
  1, 1, 0, 0,
  1, 0, 1, 0,
  0, 1, 1, 1
), byrow = TRUE, ncol = 4, nrow = 4)
power_function(A, 1000)
```

`q_analysis` *Q-analysis*

Description

Q-structure of a simplicial complex.

Usage

```
q_analysis(A, simplicial_complex = FALSE, dimensions = FALSE)
```

Arguments

`A`	An incidence matrix
`simplicial_complex`	Whether the incidence matrix is a simplices or simplicial complexes representation
`dimensions`	Return the successively chains from high to low dimensions (q) and the number of components (Q_p)

Value

This function return a q-analysis of a simplicial complex matrix

Author(s)

Alejandro Espinosa-Rada

References

Atkin, R. H. (1974). Mathematical structure in human affairs. New York: Crane, Rusak.

Freeman, L. C. (1980). Q-analysis and the structure of friendship networks. International Journal of Man-Machine Studies, 12(4), 367–378. doi:10.1016/S00207373(80)800216

Examples

```
A <- matrix(c(
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0,
  0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
  0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
  0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
  0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0
), byrow = TRUE, ncol = 19)
colnames(A) <- letters[1:ncol(A)]
rownames(A) <- 1:nrow(A)

q_analysis(A, simplicial_complex = TRUE)
```

---

recip_coef *Reciprocity*

Description

This measure calculated the reciprocity of an asymmetric matrix (directed graph).

Usage

```
recip_coef(
  A,
  diag = NULL,
  method = c("total_ratio", "ratio_nonnull", "global")
)
```

Arguments

A	A matrix
diag	Whether to consider the diagonal of the matrix
method	Whether to use `total_ratio`, `ratio_nonnull` or `global` reciprocity

Value

Return a reciprocity coefficient

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(0, 1, 1, 0,
              1, 0, 1, 0,
              0, 0, 0, 0,
              1, 0, 0, 0), byrow = TRUE, ncol = 4)
recip_coef(A)
```

---

redundancy *Redundancy measures*

Description

Redundancy measures of the structural holes theory for binary matrixes

Usage

```
redundancy(A, ego = NULL, digraph = FALSE, weighted = FALSE)
```

Arguments

A	A symmetric matrix object
ego	Name of ego in the matrix
digraph	Whether the matrix is directed or undirected
weighted	Whether the matrix is weighted or not

Value

This function returns redundancy, effective size and efficiency measures (Burt, 1992).

Author(s)

Alejandro Espinosa-Rada

References

Burt, R.S., 1992. Structural Holes: the Social Structure of Competition. Harvard University Press, Cambridge.

Borgatti, S., 1997. Unpacking Burt's redundancy measure. Connections, 20(1): 35-38.

Examples

```
A <- matrix(c(
  0, 1, 0, 0, 1, 1, 1,
  1, 0, 0, 1, 0, 0, 1,
  0, 0, 0, 0, 0, 0, 1,
  0, 1, 0, 0, 0, 0, 1,
  1, 0, 0, 0, 0, 0, 1,
  1, 0, 0, 0, 0, 0, 1,
  1, 1, 1, 1, 1, 1, 0
), ncol = 7, byrow = TRUE)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- letters[1:ncol(A)]
redundancy(A, ego = "g")
```

---

shared_partners *Shared partners*

Description

Shared partners

Usage

```
shared_partners(
  A,
  loops = FALSE,
  directed = TRUE,
  type = c("dsp", "esp", "nsp")
)
```

Arguments

`A`	A binary matrix
`loops`	Whether to consider the loops
`directed`	Whether the matrix is directed
`type`	Whether to return the `dyad-wise (dsp)` (default), `edge-wise (esp)` or `non-edgewise (nsp)` shared partners (Hunter and Handcock, 2006)

Value

This function return the distribution of shared partners.

Author(s)

Alejandro Espinosa-Rada

References

Hunter, D. R. and M. S. Handcock (2006), Inference in curved exponential family models for networks, Journal of Computational and Graphical Statistics, 15: 565– 583.

Examples

```
A <- matrix(c(
  0, 1, 0, 0, 0, 0,
  1, 0, 1, 1, 0, 1,
  0, 1, 0, 1, 0, 0,
  0, 1, 1, 0, 1, 1,
  0, 0, 0, 1, 0, 1,
  0, 1, 0, 1, 1, 0
), byrow = TRUE, ncol = 6)
shared_partners(A, type = "dsp")
shared_partners(A, type = "esp")
shared_partners(A, type = "nsp")
```

---

simplicial_complexes *Simplicial complexes*

Description

incidence matrix of simplexes or cliques

Usage

```
simplicial_complexes(A, zero_simplex = FALSE, projection = FALSE)
```

Arguments

A	A symmetric matrix object.
zero_simplex	Whether to include the zero simple.
projection	Whether to return the links between actors (i.e., rows) through their shared linking events (i.e., columns).

Value

This function return an incidence matrix of actors participating in simplices or simplicial complexes

Author(s)

Alejandro Espinosa-Rada

References

Atkin, R. H. (1974). Mathematical structure in human affairs. New York: Crane, Rusak.

Freeman, L. C. (1980). Q-analysis and the structure of friendship networks. International Journal of Man-Machine Studies, 12(4), 367–378. doi:10.1016/S00207373(80)800216

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  0, 1, 1, 0, 0, 0, 0, 1, 0,
  1, 0, 1, 0, 0, 0, 0, 0, 0,
  1, 1, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 1, 1, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 1, 1, 0,
  0, 0, 0, 0, 0, 1, 0, 1, 0,
  1, 0, 0, 0, 0, 1, 1, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0
), byrow = TRUE, ncol = 9)
```

```
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- rownames(A)
simplicial_complexes(A, zero_simplex = FALSE)
```

`spatial_cor` *Spatial autocorrelation*

Description

This function calculate some spatial autocorrelations for a sample of networks at different orders (distances).

Usage

```
spatial_cor(
  A,
  V,
  measures = c("covariance", "correlation", "moran", "geary"),
  mean = TRUE,
  diag = FALSE,
  distance1 = TRUE,
  rowstand = FALSE,
  scale = FALSE
)
```

Arguments

`A`	A symmetric matrix
`V`	A vector
`measures`	Whether to use the Covariance `covariance` (default), Correlation `correlation`, Moran I `moran` or Geary's C `geary`
`mean`	Whether to use the mean of the vector for the measures
`diag`	Whether to consider the diagonal of the matrix for the measures
`distance1`	Whether to return only the spatial autocorrelation considering the actor at distance 1
`rowstand`	Whether to use the row-standardization to estimate Moran I (Anselin, 1995)
`scale`	Whether to scale Moran I (Anselin, 1995)

Value

This function return the global spatial autocorrelation. Multiple orders can also be computed.

References

Anselin, L. (1995). Local indicators of spatial association—LISA. Geographical analysis, 27(2), 93-115.

Geary, R.C. (1954). “The Contiguity Ratio and Statistical Mapping.” The Incorporated Statistician, 5: 115-145.

Moran, P.A.P. (1950). “Notes on Continuous Stochastic Phenomena.” Biometrika, 37: 17-23.

Examples

```
A <- matrix(c(
  0, 0, 1, 1,
  0, 0, 1, 0,
  1, 0, 0, 0,
  1, 0, 1, 0
), byrow = TRUE, ncol = 4)
V <- c(2, 2, 1, 1)

spatial_cor(A, V, measures = c("moran"))
```

---

structural_na *Structural Missing Data*

Description

Assign NA to missing data in matrices.

Usage

```
structural_na(
  A,
  label = NULL,
  row_labels = NULL,
  col_labels = NULL,
  two_mode = FALSE
)
```

Arguments

A	An incident or symmetric matrix object.
label	A string vector with the names of the theoretical complete matrix (used for one-mode networks only).
row_labels	A string vector with the names of the rows (used for two-mode networks).
col_labels	A string vector with the names of the columns (used for two-mode networks).
two_mode	Boolean indicating whether the matrix is two-mode. Default is FALSE.

Value

This function returns a matrix with NA assigned to missing data.

Examples

```
# Example for one-mode network
A <- matrix(c(
  0, 1, 1,
  1, 0, 1,
  0, 0, 0
), byrow = TRUE, ncol = 3)
colnames(A) <- c("A", "C", "D")
rownames(A) <- c("A", "C", "D")
label <- c("A", "B", "C", "D", "E")
structural_na(A, label = label)

# Example for two-mode network
B <- matrix(c(
  0, 1, 0,
  1, 0, 1,
  0, 1, 0,
  1, 0, 1
), byrow = TRUE, ncol = 3)
rownames(B) <- c("X1", "X2", "X3", "X4")
colnames(B) <- c("Y1", "Y2", "Y3")
rlabels <- c("X1", "X2", "X3", "X4", "X5")
clabels <- c("Y1", "Y2", "Y3", "Y4")
structural_na(B, row_labels = rlabels, col_labels = clabels, two_mode = TRUE)
```

`struc_balance` *Structural balance*

Description

Structural balance

Usage

```
struc_balance(A, B = NULL, score = c("triangle", "walk"))
```

Arguments

A	A signed symmetric matrix (i.e., with ties that are either -1, 0 or 1)
B	A signed symmetric matrix considered as the negative ties (i.e., with ties that are either -1, 0 or 1)
score	Whether to return the `triangle` (default) or `walk` balance score (Aref and Wilson, 2017)

Value

This function return the structural balance (Heider, 1940; Cartwright and Harary, 1956). When B is used, matrix A is considered the negative matrix and A the positive matrix.

Author(s)

Alejandro Espinosa-Rada

References

Aref, Samin and Wilson, Mark C. (2017). Measuring partial balance in signed networks. Journal of Complex Networks, 6(4): 566-595.

Cartwright, Dorwin, and Harary, Frank (1956). Structural balance: a generalization of Heider's theory. Psychological review, 63(5), 277.

Heider, Fritz (1946). Attitudes and Cognitive Organization. The Journal of Psychology, 21: 107–112

Examples

```
A <- matrix(c(
  0, -1, -1, 0,
  -1, 0, 1, 0,
  -1, 1, 0, 0,
  0, 0, 0, 0
), byrow = TRUE, ncol = 4)
rownames(A) <- letters[1:nrow(A)]
colnames(A) <- rownames(A)
struc_balance(A)
```

---

trans_coef *Transitivity*

Description

This measure is sometimes called clustering coefficient.

Usage

```
trans_coef(
  A,
  method = c("weakcensus", "global", "mean", "local"),
  select = c("all", "in", "out")
)
```

Arguments

A	A matrix
method	Whether to calculate the weakcensus, global transitivity ratio, the mean transitivity or the local transitivity.
select	Whether to consider all, in or out ties for the local transitivity.

Value

Return a transitivity measure

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(c(
  0, 1, 0, 1, 0,
  1, 0, 1, 1, 0,
  0, 1, 0, 0, 0,
  1, 1, 0, 0, 1,
  0, 0, 0, 1, 0
), byrow = TRUE, ncol = 5)
rownames(A) <- letters[1:ncol(A)]
colnames(A) <- rownames(A)

trans_coef(A, method = "local")
```

`trans_matrix` *Transitivity matrix*

Description

This function assigns a one in the elements of the matrix if a group of actors are part of a transitivity structure (030T label considering the MAN triad census)

Usage

```
trans_matrix(A, loops = FALSE)
```

Arguments

`A`	A matrix
`loops`	Whether to expect nonzero elements in the diagonal of the matrix

Value

A vector assigning an id the components that each of the nodes of the matrix belongs

Author(s)

Alejandro Espinosa-Rada

References

Davis, J.A. and Leinhardt, S. (1972). "The Structure of Positive Interpersonal Relations in Small Groups." In J. Berger (Ed.), Sociological Theories in Progress, Vol. 2, 218-251. Boston: Houghton Mifflin.

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
A <- matrix(
  c(
    0, 1, 1, 0, 0, 0,
    0, 0, 1, 0, 0, 0,
    0, 0, 0, 1, 0, 0,
    0, 0, 0, 0, 0, 0,
    0, 0, 1, 1, 0, 0,
    0, 0, 0, 0, 0, 0
  ),
  byrow = TRUE, ncol = 6
)
rownames(A) <- letters[1:NROW(A)]
colnames(A) <- rownames(A)
trans_matrix(A, loops = TRUE)
```

`triad_uman` *Triad census analysis assuming U|MAN*

Description

Considering the triad census of Davis and Leinhardt (1972) for vector A, B, and C:

Usage

```
triad_uman(A, ztest = FALSE, covar = FALSE)
```

Arguments

`A`	A symmetric matrix object
`ztest`	Return Z and p-value
`covar`	Return the covariance matrix for triadic analysis

Details

003 = A,B,C, empty triad

012 = A -> B, C, triad with a single directed edge

102 = A <-> B, C, triad with a reciprocated connection between two vertices

021D = A <-B-> C, triadic out-star

021U = A -> B <- C triadic in-star

021C = A-> B-> C, directed line

111D = A <-> B <-C

111U = A <-> B-> C

030T = A-> B <-C, A-> C

030C = A <-B <-C, A-> C

201 = A <-> B <-> C

120D = A <-B-> C, A <-> C

120U = A-> B <-C, A <->C

120C = A-> B-> C, A <-> C

210 = A-> B <-> C, A <-> C

300 = A <-> B <-> C, A <->C, complete triad.

Value

This function gives the counts of the triad census, the expected counts, assuming that U|MAN distribution (Holland and Leinhardt, 1975, 1976) is operating, and the standard deviations of these counts.

Author(s)

Alejandro Espinosa-Rada

References

Davis, J.A. and Leinhardt, S. (1972). The Structure of Positive Interpersonal Relations in Small Groups. In J. Berger (Ed.), Sociological Theories in Progress, Volume 2, 218-251. Boston: Houghton Mifflin.

Holland, P. W. and Leinhardt, S. (1975). The statistical analysis of local structure in social networks. In D. R. Heise (Ed.), Sociological Methodology, 1976 (Jossey-Bass, pp. 1–45).

Holland, P. W. and Leinhardt, S. (1976). Local Structure in Social Networks. Sociological Methodology, 7, 1–45. doi:10.2307/270703

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
data(krackhardt_friends)
triad_uman(krackhardt_friends)

triad_uman(krackhardt_friends, ztest = TRUE, covar = TRUE)
```

---

zone_sample *Zone-2 sampling from second-mode*

Description

Second-zone multilevel sampling considering a second-mode focal actor

Usage

```
zone_sample(A, X, ego = TRUE, core = FALSE)
```

Arguments

A	A symmetric matrix object.
X	X an incidence matrix object.
ego	Whether to add or not ego into the subgraph.
core	Whether to add actors at distance one from ego

Value

This function return a list of second-zone subgraphs using as a focal actor the second-mode of the multilevel network.

Author(s)

Alejandro Espinosa-Rada

References

Espinosa-Rada, A. (2021). A Network Approach for the Sociological Study of Science: Modelling Dynamic Multilevel Networks. [PhD](https://research.manchester.ac.uk/en/studentTheses/a-network-approach-for-the-sociological-study-of-science-and-know). The University of Manchester.

Examples

```
A <- matrix(c(
  0, 1, 0, 0, 0, 0, 0, 0,
  0, 0, 1, 0, 0, 0, 0, 0,
  0, 1, 0, 1, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0
), byrow = TRUE, ncol = 8)
colnames(A) <- c("1", "2", "3", "4", "5", "6", "7", "8")
rownames(A) <- c("1", "2", "3", "4", "5", "6", "7", "8")

X <- matrix(c(
  1, 0, 0, 0,
  1, 0, 0, 0,
  1, 0, 1, 0,
  0, 1, 1, 0,
  0, 1, 1, 1,
  0, 1, 0, 0,
  0, 0, 0, 0,
  0, 0, 0, 1
), byrow = TRUE, ncol = 4)
colnames(X) <- c("a", "b", "c", "d")
rownames(X) <- c("1", "2", "3", "4", "5", "6", "7", "8")

set.seed(18051889)
zone_sample(A, X, core = TRUE)
```

`z_arctest` *Z test of the number of arcs*

Description

Z test of the number of arcs

Usage

```
z_arctest(G, p = 0.5, interval = FALSE)
```

Arguments

G	A symmetric matrix object.
p	Constant probability p.
interval	Return a 95 percent confidence interval.

Value

This function gives a Z test and p-value for the number of lines or arcs present in a directed graph

Author(s)

Alejandro Espinosa-Rada

References

Wasserman, S. and Faust, K. (1994). Social network analysis: Methods and applications. Cambridge University Press.

Examples

```
data(krackhardt_friends)
z_arctest(krackhardt_friends)
```

Index