

# Package: isnar (via r-universe)

September 6, 2024

**Version** 1.0-0

**License** GPL-2

**Title** Introduction to Social Network Analysis with R

**Description** Functions and datasets accompanying the workshop  
``Introduction to Social Network Analysis with R" on annual  
INSNA Sunbelt conferences.

**Depends** R (>= 2.10)

**Imports** igraph (>= 0.6-0), mouter

**Suggests** testthat, roxygen2

**RoxygenNote** 6.1.1

**LazyLoad** yes

**LazyData** yes

**Encoding** UTF-8

**Remotes** mbojan/mouter

**Repository** <https://mbojan.r-universe.dev>

**RemoteUrl** <https://github.com/mbojan/isnar>

**RemoteRef** HEAD

**RemoteSha** 56177701d509b267eff845e15514e4cf48e69675

## Contents

assort . . . . .	2
Catania . . . . .	4
coauthorship . . . . .	5
coleman . . . . .	5
earnings . . . . .	6
EFnet . . . . .	7
ei . . . . .	7
fold . . . . .	9
freeman . . . . .	10
full_mm . . . . .	11

Galesburg2 . . . . .	13
gamix . . . . .	14
gender_roles . . . . .	15
group_sizes . . . . .	15
IBE121 . . . . .	16
Input_output . . . . .	17
isnar . . . . .	17
judge_net . . . . .	18
judge_net_bp . . . . .	18
mixingm . . . . .	19
orwg . . . . .	20
pairwise_redundancy . . . . .	21
read_highland_tribes . . . . .	22
smi . . . . .	23
ssi . . . . .	24
symmetrize . . . . .	26
Wnet . . . . .	27

<b>Index</b>	<b>29</b>
--------------	-----------

---

assort	<i>Assortativity Coefficient</i>
--------	----------------------------------

---

### Description

Assortativity coefficient is a measure of segregation for social networks due to Mark Newman (2002).

### Usage

```
assort(object, ...)

## S3 method for class 'table'
assort(object, ...)

## S3 method for class 'igraph'
assort(object, vattr, ...)

## Default S3 method:
assort(object, ...)
```

### Arguments

object	R object, see available methods
...	other arguments to/from other methods
vattr	character, name of the vertex attribute for which the measure is to be calculated

## Details

The measure evaluates the relative prevalence of within-group ties. It is based on the contact layer of the mixing matrix.

Assortativity coefficient is 1 if all ties are within-group. The minimum can be negative, but not less than -1, and depends on the relative number of ties of nodes in different groups. If the network conforms to "proportionate mixing", the coefficient is 0.

If object is a table it is treated as a mixing matrix. Two-dimensional table is interpreted as a contact layer. Three-dimensional table is interpreted as a full mixing matrix  $m_{ghy}$  cross-classifying all dyads, in which 'g' and 'h' correspond to group membership of ego and alter respectively. Layers  $y=1$  and  $y=2$  are assumed to be non-contact and contact layers respectively. In the 3-d case only  $g[, , 2]$  is used.

If  $g$  is an object of class "igraph" the measure is calculated for the vertex attribute specified with `vattr`.

For any other classes, object  $g$  are coerced to a table and the table method is called.

## Value

Numeric value of the index.

## References

Newman, M. J. and Girvan, M. (2002) "Mixing patterns and community structure in networks", arXiv:cond-mat/0210146v1

Newman, M. J. (2003) "Mixing patterns in networks" arXiv:cond-mat/0209450v2

## See Also

Mixing matrices: [mixingm](#)

Other segregation measures: [coleman](#), [ei](#), [freeman](#), [gamix](#), [orwg](#), [smi](#), [ssi](#)

## Examples

```
assort(Wnet, "gender")
assort(EFnet, "type")

if( require(igraph, quietly = TRUE) ) {
  # value of 'assort' for full networks of different sizes
  f <- function(n)
  {
    gfull <- graph.full(n, directed=FALSE)
    V(gfull)$type <- rep(1:2, length=vcount(gfull))
    assort(gfull, "type")
  }
  set.seed(1)
  x <- sort(sample(5:100, 25) * 2)
  y <- sapply(x, f)
  plot(x, y, type="o",
```

```

xlab="Network size", ylab="Assortativity coefficient",
main="Assortativity coef. for full networks of different sizes")
}

```

---

Catania

*Pattern of sexual contacts in AMEN study*

---

### Description

Contact layer of the mixing matrix of men and women in US based on "Aids in Multi-Ethnic Neighborhoods" (AMEN). Based on Newman (2003).

### Format

Four-by-four numeric matrix with dimnames.

	woman			
man	black	hispanic	white	other
black	0.258	0.016	0.035	0.013
hispanic	0.012	0.157	0.058	0.019
white	0.013	0.023	0.306	0.035
other	0.005	0.007	0.024	0.016

### Source

Newman, M. (2003) "Mixing patterns in networks" Arxiv:cond-mat/0209450 v2

### References

Catania et al. (1992) "The population-based AMEN (AIDS in Multi-Ethnic Neighborhoods) study" American Journal of Public Health 82, 284-287

Morris, M. (1995) "Data driven network models for the spread of infectious disease". In D. Mollison (ed.) "Epidemic Models: Their Structure and Relation to Data", pp. 302-322, Cambridge University Press, Cambridge

Newman, M. (2003) "Mixing patterns in networks" Arxiv:cond-mat/0209450 v2

### Examples

```

data(Catania)

# assortativity
ep <- sum(Catania %% Catania)
( sum(diag(Catania)) - ep ) / ( 1 - ep )

```

---

coauthorship	<i>Coauthorship network</i>
--------------	-----------------------------

---

**Description**

Coauthorship network of scientists from University of Warsaw and their external co-workers. Relation indicates pairs of people who have authored at least one publication together.

**Format**

Object of class `igraph` of size 10114, undirected.

**Details**

Node attributes include authors' direct affiliation and also their affiliation to faculty.

**Source**

Own calculation

---

coleman	<i>Coleman's homophily index</i>
---------	----------------------------------

---

**Description**

Coleman's homophily index for directed networks.

**Usage**

```
coleman(object, ...)
```

```
## S3 method for class 'table'
coleman(object, gsizes = NULL, loops = FALSE, ...)
```

```
## S3 method for class 'igraph'
coleman(object, vattr, ...)
```

```
## Default S3 method:
coleman(object, ...)
```

**Arguments**

<code>object</code>	R object, see Details for available methods
<code>...</code>	other arguments passed to/from methods
<code>gsizes</code>	numeric vector of group sizes
<code>loops</code>	logical, whether loops are allowed
<code>vattr</code>	character, vertex attribute

**Details**

Coleman's homophily index computes homophily scores for each group defined by a vertex attribute.

If object is a table it is interpreted as a mixing matrix. If it is only the contact layer (2-dimensional), then vector of group sizes need to be supplied via `gsizes`.

object can be of class "igraph"

Default method tries to coerce object to table and use other methods.

**Value**

Vector of numeric values of the index for each group

**References**

Coleman, J. (1958) "Relational analysis: The study of social organizations with survey methods", *Human Organization* 17:28–36.

**See Also**

Other segregation measures: [assort](#), [ei](#), [freeman](#), [gamix](#), [orwg](#), [smi](#), [ssi](#)

**Examples**

```
if( require(igraph, quietly = TRUE)) {
  coleman(as.directed(Wnet, "mutual"), "gender")
  coleman(as.directed(EFnet, "mutual"), "type")
}
```

---

earnings

*Subset of Polish General Social Survey data with data on incomes*

---

**Description**

Subset of Polish General Social Survey data with data on incomes

**Format**

SPSS system file

**Source**

Polish Social Science Data Archive (ADS)

**References**

PGSS reference

---

EFnet

*Example data from Echenique and Fryer (2006)*

---

### Description

Artificial example data from Echenique & Fryer (2006) representing a city with black and white neighbourhoods.

### Format

Object of class "igraph". An undirected network with vertex attributes

- type (values 1 or 2).
- name

### Details

This data is taken from Echenique & Fryer (2006, figure III). The data represent a fictional city composed of 30 neighborhoods that are either black or white.

### Source

Echenique, Federico and Roland G. Fryer, Jr. (2006) "A Measure of Segregation Based On Social Interactions" *Quarterly Journal of Economics* CXXII(2):441-485

### Examples

```
if( require(igraph, quietly = TRUE) ) {  
  set.seed(2992)  
  plot(EFnet, layout=layout.fruchterman.reingold,  
        vertex.color=V(EFnet)$type+1, vertex.label.family="",  
        sub="Source: Echenique & Fryer (2006)",  
        main="Neighborhood racial segregation\n in a fictional city" )  
}
```

---

ei

*Krackhard and Stern's E-I index*

---

### Description

An index proposed by Krackhard and Stern (1988) to capture relative prevalence of between- and within-group ties. From that perspective it can be interpreted as a measure of network segregation.

## Usage

```
ei(object, ...)  
  
## S3 method for class 'table'  
ei(object, ...)  
  
## S3 method for class 'igraph'  
ei(object, vattr, directed = is.directed(object),  
    loops = any(is.loop(object)), ...)  
  
## Default S3 method:  
ei(object, ...)
```

## Arguments

object	R object, see Details for available methods
...	other arguments passed to/from other methods
vattr	character scalar or vector of length equal to the size of object, vertex attribute for which mixing matrix is to be computed
directed	logical whether the network is directed
loops	logical, whether loops are allowed

## Details

If object is a table, it is assumed to be a mixing matrix.

Method for igraphs

Default method tries to coerce object to table.

## Value

Numerical value of the E-I index.

## See Also

Other segregation measures: [assort](#), [coleman](#), [freeman](#), [gamix](#), [orwg](#), [smi](#), [ssi](#)

## Examples

```
ei(Wnet, "gender")
```



---

`fold`*Folding square matrices around the diagonal*

---

**Description**

Fold a square matrix by collapsing lower triangle on upper triangle, or vice versa, through addition.

**Usage**

```
fold(x, direction = c("upper", "lower"))
```

**Arguments**

<code>x</code>	square numeric matrix
<code>direction</code>	character, one of "upper" or "lower", direction of folding

**Details**

By default, for `direction=="upper"`, the function takes the values in the lower triangle of `x` and adds them symmetrically to the values in the upper triangle. The values on the diagonal remain unchanged. The lower triangle is filled with 0s. If `direction=="lower"` the upper triangle is collapsed on to the lower triangle.

**Value**

Square matrix of the same dim as `x` with the lower (upper) triangle folded onto the upper (lower) triangle.

**See Also**

[upper.tri](#), [lower.tri](#)

**Examples**

```
(m <- matrix(1:4, 2, 2))
(f1 <- fold(m))
(f2 <- fold(m, "lower"))

stopifnot( all.equal(diag(m), diag(f1)) )
stopifnot( all.equal(diag(m), diag(f2)) )
stopifnot( all.equal(f1[1,2], m[2,1] + m[1,2]) )
stopifnot( all.equal(f2[2,1], m[2,1] + m[1,2]) )
```

freeman

*Generalized Freeman's segregation index***Description**

Calculate Freeman's segregation index for undirected networks with arbitrary number of groups.

**Usage**

```
freeman(object, ...)

## S3 method for class 'table'
freeman(object, gsizes = NULL, more = FALSE,
        loops = FALSE, ...)

## S3 method for class 'igraph'
freeman(object, vattr, gsizes = NULL,
        loops = any(is.loop(object)), ...)

## Default S3 method:
freeman(object, ...)
```

**Arguments**

object	R object, see Details for available methods
...	other arguments passed to/from other methods
gsizes	numeric, optional true distribution of types, see Details
more	logical, should some more output be returned
loops	logical, whether loops are allowed
vattr	character scalar or any vector of length equal to <code>vcount(object)</code> , name of the vertex attribute in object designating the groups or a vector with the attribute itself

**Details**

Freeman's segregation index (Freeman, 1978) is designed to capture the extent to which the defined groups of vertices tend to have more edges with vertices from the same group than with other groups. Formally, the index compares the observed number of between-group ties with the number of between-group ties that would be expected if ties would be created randomly.

The index has some discontinuity as there are network and group configurations that are characterized by the higher number of between-group ties that is expected under a random graph. The index truncates these situations and takes a value of 0.

The original Freeman's formulation was for only two types of vertices. Here it is extended to the arbitrary number of types. The modification only affects the way in which the expected number of inter-type edges under pure random graph is calculated.

The function internally calculates the frequency of types of vertices in the supplied attributer `vattr`. However, it is possible to override this by specifying “true” type distribution with the `dis` argument. It is assumed to be a table (as returned by `table`) or a numeric vector with frequencies of types of vertices. This may be especially usefull when dealing with large graphs with larger number of isolates.

Method for mixing matrices

Method for "igraph"s

### Value

The value of the Freeman’s index.

If `more` is TRUE, some intermediate results are returned in a list.

### References

Freeman, Linton C. (1978) Segregation in Social Networks, *Sociological Methods & Research* 6(4):411–429

### See Also

Other segregation measures: [assort](#), [coleman](#), [ei](#), [gamix](#), [orwg](#), [smi](#), [ssi](#)

### Examples

```
## White's data from Freeman's article
# segregation level
freeman(Wnet, "gender")
# using 'more' argument
freeman(Wnet, "gender", more=TRUE)
```

---

full\_mm

*Create a 3-dimensional mixingm matrix*

---

### Description

Given contact layer of the mixing matrix compute a full 3-dimensional mixing matrix by rebuilding the non-contact layer from supplied group sizes and other arguments.

### Usage

```
full_mm(cl, gsizes, directed = TRUE, loops = FALSE)
```

**Arguments**

<code>c1</code>	numeric matrix with contact layer of the mixing matrix
<code>gsizes</code>	numeric vector or matrix with group sizes, see Details.
<code>directed</code>	logical, whether the network is directed
<code>loops</code>	logical, whether loops (self-ties) are allowed

Contact layer of the mixing matrix is a cross-classification of ties according to the attributes of tie sender (ego) and tie receiver (alter). Classically, the same attribute is used for ego and alter resulting in a square mixing matrix. In such cases `c1` should be square with `c1[i, j]` being the number of ties from actors in group `i` to actors in group `j` and `gsizes` should be a numeric vector with number of nodes in each group. Consequently, it must hold that `length(gsizes) == nrow(c1) == ncol(c1)`.

In more general case we can use different node attributes for ego and alter. Then `c1` does not have to be square. In such cases `gsizes` should be a cross-tabulation of nodes according to their values on both attributes. See Examples.

**Value**

`full_mm` returns a full three-dimensional mixing matrix as an array with `dim` attribute equal to `c(nrow(c1), ncol(c1), 2)`.

**Examples**

```
### Square example

# Contact layer of the mixing matrix
mm1 <- matrix( c( 20, 10, 5,
                 12, 30, 10,
                 3, 11, 25 ),
              byrow=TRUE, ncol=3, nrow=3)
dimnames(mm1) <- list(ego=letters[1:3], alter=letters[1:3])
mm1

# Assuming some group sizes
gs1 <- c(a=9, b=12, c=10)

# Full mixing matrix
full_mm( mm1, gs1)

### Non-square example

# Mixing matrix
# Now using different attributes for ego and alter
mm2 <- cbind(mm1, c(20, 10, 5))
colnames(mm2) <- LETTERS[1:4]
names(dimnames(mm2)) <- c("ego", "alter")
```

```

mm2

# Create artificial distribution of attributes
set.seed(123)
a1 <- sample(letters[1:3], sum(gs1), replace=TRUE, prob=gs1/sum(gs1))
table(a1)
a2 <- sample(LETTERS[1:4], sum(gs1), replace=TRUE)
table(a2)
(x <- table(a1, a2))      # Cross-tabulation

# Full mixing matrix
full_mm( mm2, gsizes=x)

```

---

Galesburg2

*Columbia University Drug Study data 2*


---

### Description

Data from Columbia University Drug Study on diffusion of medical innovation in the form of a new drug (gammanym). Its a network of 31 medical doctors connected with friendship or discussion links. Vertex attribute "adoption" specifies time of first prescription of the new drug.

### Format

Directed network (class 'igraph') with edge attributes:

**discussion** logical, discussion link

**friendship** logical, friendship nomination

and vertex attributes:

**adoption** numeric, time of adoption

### Source

Pajek datasets <http://vlado.fmf.uni-lj.si/pub/networks/data/esna/Galesburg2.htm>

### References

Coleman, J.S., E. Katz, H. Menzel (1966) "Medical Innovation. A Diffusion Study", Indianapolis: Bobbs-Merrill

### Examples

```

if(require(igraph, quietly = TRUE)) {
  plot(Galesburg2, edge.arrow.size=.3)
}

```

gamix

*Gupta-Anderson-May measure of within-group mixing***Description**

Measure of within-group mixing in networks proposed in Gupta, Anderson and May (1989).

**Usage**

```
gamix(object, ...)

## S3 method for class 'table'
gamix(object, debug = FALSE, ...)

## S3 method for class 'igraph'
gamix(object, vattr, ...)

## Default S3 method:
gamix(object, ...)
```

**Arguments**

object	R object, see Details for available methods
...	other objects passed to/from other methods
debug	logical, return some intermediate results as attributes to the returned value
vattr	character, name of vertex attribute

**Details**

The measure varies between  $-1/\text{vcount}(g)$  for disassortative mixing and 1 for perfect within-group mixing. It takes a value of 0 for proportionate mixing.

Method for mixing matrices

Method for igraphs

**Value**

Numerical value of the measure.

**References**

Gupta, S., Anderson, R., May, R. (1989) "Networks of sexual contacts: implications for the pattern of spread of HIV", AIDS 3:807–817

**See Also**

Other segregation measures: [assort](#), [coleman](#), [ei](#), [freeman](#), [orwg](#), [smi](#), [ssi](#)

**Examples**

```
gamix(Wnet, "gender")
gamix(EFnet, "type")
```

---

gender_roles	<i>Subset of Polish General Social Survey data with response to questions on gender roles</i>
--------------	---

---

**Description**

Subset of Polish General Social Survey data with response to questions on gender roles

**Format**

Plain text file with header, space-separated values, and the following columns

**Source**

Polish Social Science Data Archive (ADS)

**References**

PGSS reference

---

group_sizes	<i>Computing group sizes from square mixing matrices group_sizes recomputes group sizes from a full mixing matrix. This is only limited to square (single-attribute) mixing matrices.</i>
-------------	---

---

**Description**

Computing group sizes from square mixing matrices group\_sizes recomputes group sizes from a full mixing matrix. This is only limited to square (single-attribute) mixing matrices.

**Usage**

```
group_sizes(mm, directed = TRUE, loops = FALSE)
```

**Arguments**

mm	numeric array with dim of (k, k, 2) for some 'k'
directed	logical, whether network is directed
loops	logical, whether loops are allowed

**Value**

A numeric vector of group sizes

**Examples**

```
# For White's data
mm <- mixingm( Wnet, "gender", full=TRUE )
```

---

IBE121

*Classroom network*

---

**Description**

Classroom network (directed multigraph) based on a study by (Polish) Educational Research Institute. Relations come from sociometric questions (loose translation from Polish):

1. With whom would you like to play with?
2. With whom would you share a secret?
3. Imagine you are to work in groups during class. With whom would you like to work in a group?
4. With whom would you rather not play?

Node attributes include gender, IQ score (Raven's test), socioeconomic position of parents.

**Format**

Object of class `igraph` with a directed multigraph (possible multiple edges within the same pair of vertices) of size 26. Edge attributes:

**female** gender

**Source**

Educational Research Institute

**References**

Educational Research Institute



---

Input\_output

*Input-output table for U.S. economy*

---

**Description**

Based on data from TODO

**Format**

22-by-22 numeric matrix with dimnames.

**Source**

TODO

**References**

TODO: Bojanowski () "Industrial structure and interfirm collaboration".

---

isnar

*Introduction to Social Network Analysis with R*

---

**Description**

Functions and data accompanying my workshop "Introduction to Social Network Analysis with R".

**Terminological disclaimer**

This package is about analyzing social networks. Social Network Analysis (SNA) uses mathematical graph theory to represent its ideas. Nevertheless, for, among other things, historical reasons SNA developed its own terminology of networks that functions somewhat in parallel to graph-theoretical terms. To clarify, these are the synonyms for strict mathematical terms corresponding to graphs:

**graph** a collection of vertices and lines connecting them. Sometimes called "network"

**vertex** sometimes called "node"

**line, edge, arc** sometimes called a (directed/undirected) "tie". Edge starting point can be called a "tie sender" or "ego". Edge endpoint can be called "tie receiver" or "alter"

---

judge\_net

*Judges network*

---

**Description**

Network of judges from one of the Polish regional courts. Relation indicates which judges have ruled in at least one case together. This network is a projection from bipartite network [judge\\_net\\_bp](#).

**Format**

Object of class igraph of size 40, undirected, with predefined layout.

**Details**

Node attributes include gender and code of division.

**Source**

Own calculation based on [SAOS](#)

---

judge\_net\_bp

*Bipartite judges–judgments network*

---

**Description**

Two-mode network with classes representing judges and judgments from one of the Polish regional courts. Relation indicates which judges were involved in each case.

**Format**

Object of class igraph of size 1189, undirected, bipartite.

**Details**

Node attributes include judges' gender and code of division. Attribute 'type' indicates classes of nodes, in accordance with igraph representation of bipartite networks, TRUE for judges and FALSE for judgments.

**Source**

Own calculation based on [SAOS](#)

---

mixingm	<i>Network mixing matrix</i>
---------	------------------------------

---

## Description

Creating network mixing matrices.

## Usage

```
mixingm(object, ...)
```

```
## S3 method for class 'igraph'
mixingm(object, rattr, cattr = rattr, full = FALSE,
        directed = is.directed(object), loops = any(is.loop(object)), ...)
```

## Arguments

object	R object, see Details for available methods
...	other arguments passed to/from other methods
rattr	name of the vertex attribute or an attribute itself as a vector. If cattr is not NULL, rattr is used for rows of the resulting mixing matrix.
cattr	name of the vertex attribute or an attribute itself as a vector. If supplied, used for columns in the mixing matrix.
full	logical, whether two- or three-dimensional mixing matrix should be returned.
directed	logical, whether the network is directed. By default, directedness of the network is determined with <code>is.directed</code> .
loops	logical, whether loops are allowed. By default it is TRUE whenever there is at least one loop in object.

## Details

Network mixing matrix is, traditionally, a two-dimensional cross-classification of edges depending on the values of a specified vertex attribute for tie sender and tie receiver. It is an important tool for assessing network homophily or segregation.

Let  $G$  be the number of distinct values of the vertex attribute in question. We may say that we have  $G$  mutually exclusive groups in the network. The mixing matrix is a  $G \times G$  matrix such that  $m_{ij}$  is the number of ties send by vertices in group  $i$  to vertices in group  $j$ . The diagonal of that matrix is of special interest as, say,  $m_{ii}$  is the number of ties *within* group  $i$ .

A full mixing matrix is a three-dimensional array that cross-classifies *all* network *dyads* depending on:

1. the value of the vertex attribute for tie sender
2. the value of the vertex attribute for tie receiver
3. the status of the dyad, i.e. whether it is connected or not

The two-dimensional version is a so-called "contact layer" of the three-dimensional version.

If object is of class "igraph," mixing matrix is created for the network in object based on vertex attributes supplied in arguments `rattr` and optionally `cattr`.

If only `rattr` is specified (or, equivalently, `rattr` and `cattr` are identical), the result will be a mixing matrix  $G \times G$  if `full` is FALSE or  $G \times G \times 2$  if `full` is TRUE. Where  $G$  is the number of categories of vertex attribute specified by `rattr`.

If `rattr` and `cattr` can be used to specify different vertex attributes for the sender and the receiver.

### Value

Depending on `full` argument a two- or three-dimensional array crossclassifying connected or all dyads in object.

For undirected network and if `foldit` is TRUE (default), the matrix is folded onto the upper triangle (entries in lower triangle are 0).

### Examples

```
if(require(igraph, quietly = TRUE)) {
  # some directed network
  net <- graph(c(1,2, 1,3, 2,3, 4,5, 1,4, 1,5, 4,2, 5,3))
  V(net)$type <- c(1,1,1, 2,2)
  mixingm(net, "type")
  mixingm(net, "type", full=TRUE)
  # as undirected
  mixingm(as.undirected(net), "type")
  mixingm(net, "type")
  mixingm(net, "type", full=TRUE)
}
```

---

orwg

*Odds ratio of existence of within-group ties*

---

### Description

Odds ratio for connected, as opposed to disconnected, dyads depending whether it is between- or within-group, i.e. how much more likely the dyad will be connected if it is within-group.

### Usage

```
orwg(object, ...)

## S3 method for class 'table'
orwg(object, ...)

## S3 method for class 'igraph'
orwg(object, vattr, ...)

## Default S3 method:
orwg(object, ...)
```

**Arguments**

object	R object, see Details for available methods
...	other arguments passed to/from other methods
vattr	character scalar or any vector, name of the vertex attribute or the attribute itself (as a vector)

**Details**

The measure takes values, like all odds ratios, from (0; Inf).

Method for mixing matrices.

Method for igraphs

**Value**

Numeric value of the measure.

**References**

Moody, Jim (2001) "Race, school integration, and friendship segregation in America", *American Journal of Sociology*, 107(3):679–377

**See Also**

Other segregation measures: [assort](#), [coleman](#), [ei](#), [freeman](#), [gamix](#), [smi](#), [ssi](#)

**Examples**

```
orwg(Wnet, "gender")
```

---

`pairwise_redundancy`    *Pairwise tie redundancy*

---

**Description**

Measuring relative redundancy of ties.

**Usage**

```
pairwise_redundancy(g, v = igraph::V(g))
```

```
pr_invdistance(g, v = igraph::V(g))
```

```
pr_sharedp(g, v = igraph::V(g))
```

**Arguments**

<code>g</code>	graph
<code>v</code>	nodes to be treated as egos

**Details**

`pairwise_redundancy` computes a combination of the two measures:

**Value**

Every function returns a data frame with columns `ego`, `v1`, `v2` with codes for, respectively, `ego`, `alter1`, and `alter2`. Additional column depends on function and it is:

**invdis** inverse distance returned by `pr_invdistance`

**sharedp** number of shared partners returned by `pr_sharedp`

**pr\_redundancy** the two above combined returned by `pairwise_redundancy`

---

`read_highland_tribes` *Read Highland Tribes*

---

**Description**

Hage & Harary (1983) use the Gahuku-Gama system of the Eastern Central Highlands of New Guinea, described by Read (1954), to illustrate a clusterable signed graph. Read's ethnography portrayed an alliance structure among three tribal groups containing balance as a special case; among Gahuku-Gama the enemy of an enemy can be either a friend or an enemy.

**Usage**

```
read_highland_tribes
```

**Format**

Igraph object with undirected network with 16 vertices. Vertices have original names of the tribes (vertex attribute name). Edge attributes

**positive** logical, whether a tie is positive or negative

**Source**

UCINET IV datasets retrieved from Pajek data collection <http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm#gama>

**References**

Hage P. and Harary F. (1983). Structural models in anthropology. Cambridge: Cambridge University Press. (See p 56-60).

Read K. (1954). Cultures of the central highlands, New Guinea. Southwestern Journal of Anthropology, 10, 1-43.

**Examples**

```

if( require(igraph, quietly=TRUE) ) {
plot(read_highland_tribes,
  vertex.color=ifelse(E(read_highland_tribes)$positive, "black", "red"),
  layout=layout.fruchterman.reingold(delete.edges(read_highland_tribes,
    E(read_highland_tribes)[!positive])),
  edge.curved=0.1)
}

```

smi

*Segregation Matrix Index***Description**

Segregation Matrix Index due to Freshman (1997). A measure of network segregation. Currently (and originally) supports only two groups.

**Usage**

```

smi(object, ...)

## S3 method for class 'table'
smi(object, normalize = TRUE, ...)

## S3 method for class 'igraph'
smi(object, vattr, ...)

## Default S3 method:
smi(object, ...)

```

**Arguments**

object	R object, see Details for available methods
...	other arguments passed to/from other methods
normalize	logical, whether normalized values should be returned, defaults to TRUE
vattr	character, name of the node attribute designating groups

**Details**

The Segregation Matrix Index (SMI) is calculated for every group separately. It compares the density within group to the density of between group ties of nodes belonging to that group.

Non-normalized version is the ratio of the within-group density to the between-group density, so vary between 0 and infinity. The normalized version varies between 0 and 1.

Method for mixing matrices.

Method for igraphs

**Value**

Numeric vector of length equal to the number of groups in `g` according to `vattr` with the values of SMI for the groups.

**References**

Freshman, M. (1997) "Cohesive Group Segregation Detection in a Social Network by the Segregation Matrix Index", *Social Networks*, 19:193–207

**See Also**

Other segregation measures: [assort](#), [coleman](#), [ei](#), [freeman](#), [gamix](#), [orwg](#), [ssi](#)

**Examples**

```
if(require(igraph, quietly = TRUE)) {
  data(Wnet)
  smi( as.directed(Wnet, "mutual"), "gender")
}
```

---

 ssi

*Spectral Segregation Index for Social Networks*


---

**Description**

These functions implement Spectral Segregation Index as proposed by Echenique & Fryer (2006). This index is a node-level measure of segregation in a given network.

**Usage**

```
ssi(g, vattr)
```

**Arguments**

<code>g</code>	object of class "igraph" representing a network
<code>vattr</code>	character, name of the vertex attribute

**Details**

For a full description and axiomatization see Echenique & Fryer (2006).

The network `g` is converted to adjacency matrix and normalized so that all rows sum-up to 1.

The procedure essentially consists of creating a submatrix, say,  $B$  of the adjacency matrix, say  $A$ . This submatrix  $B$  contains only vertices of the given type. It may be viewed as a type-homogeneous subnetwork of  $A$ . This subnetwork is further decomposed into connected components. Then, for every component, an eigenvalue decomposition is applied. The value of the index for the component is simply the largest eigenvalue, and the individual-level indices are obtained by distributing it according to the corresponding eigenvector.



**Value**

Named vector of individual level values of SSI. Names correspond to vertex ids in g.

**References**

Echenique, Federico and Roland G. Fryer, Jr. (2006) A Measure of Segregation Based On Social Interactions

**See Also**

Other segregation measures: [assort](#), [coleman](#), [ei](#), [freeman](#), [gamix](#), [orwg](#), [smi](#)

**Examples**

```
if( require(igraph, quietly = TRUE)) {

  ### artificial EF data
  x <- ssi(EFnet, "type")
  x

  # show it on picture
  a <- V(EFnet)$type
  # rescale SSI values to use as shades of gray
  k <- 1 - scale(x, center=min(x), scale=max(x) - min(x))
  plot( EFnet, layout=layout.fruchterman.reingold,
        vertex.color= gray(k),
        vertex.label.family="",
        vertex.shape=c("circle", "square")[a],
        vertex.label.color=gray( (1-k) > .4 ) )

  ### For White's kinship data
  x <- ssi(Wnet, "gender")
  x
  # plot it
  a <- V(Wnet)$gender
  k <- 1 - scale(x, center=min(x), scale=max(x) - min(x))
  set.seed(1234)
  plot( Wnet, layout=layout.fruchterman.reingold,
        main="Node segregation in White's kinship data",
        vertex.label.family="",
        vertex.label=V(Wnet)$name,
        vertex.color= gray(k),
        vertex.shape=c("circle", "square")[a],
        vertex.label.color="black")
  legend( "topleft", legend=c("Men", "Women"), pch=c(0,1), col=1)

}
```

---

symmetrize                      *(De)symmetrize square numeric matrix*

---

### Description

(De)symmetrize square binary matrix in various ways.

### Usage

```
symmetrize(mat, rule = c("upper", "lower", "div", "intdiv"))
```

### Arguments

mat	square numeric matrix
rule	character, direction of copying, see Details

### Details

Argument `mat` is to be a square numeric matrix. The way it is made symmetric, or asymmetric, depends on the value of the `rule` argument.

If `rule` is "upper" or "lower" then `mat` is made symmetric by copying, respectively, upper triangle onto lower, or lower onto upper. The value of `rule` specifies values of which triangle will stay in the returned value.

If `rule` is "intdiv" then the off-diagonal values are distributed approximately equally between the lower/upper triangles. If `r` is the computed result, then `r[i, j]` will be equal to  $(x[i, j] + x[j, i]) / 2$  if `r[i, j]` is in the lower triangle. It will be equal to  $(x[i, j] + x[j, i]) / (2 + 1)$  if in the upper triangle.

If `rule` is "div" then the off-diagonal values are distributed equally between the lower/upper triangles: as with "intdiv" but using normal / division.

### Value

A matrix: symmetrized version of `mat`.

### See Also

[fold](#)

### Examples

```
m <- matrix(1:16, 4, 4)

# copy upper triangle onto lower symmetrically
symmetrize(m, "upper")

# copy lower triangle onto upper symmetrically
symmetrize(m, "lower")
```

```
# distribute off-diagonal values exactly
# r[i,j] = (m[i,j] + m[j,i]) / 2
r1 <- symmetrize(m, "div")
r1
all.equal(sum(m), sum(r1))

# distribute off-diagonal values using integer division
r2 <- symmetrize(m, "intdiv")
r2
all.equal(sum(m), sum(r2))
```

---

Wnet

*White's data on Effective Kinship Networks*

---

### Description

This data is taken from Freeman (1978) who uses data from White (1975) to illustrate the segregation measure.

### Format

Object of class "igraph" with an undirected network of size 10. Vertex attribute gender, takes values 1=woman, 2=man.

### Details

Based on Freeman (1978):

White dealt with the problem of segregation among social positions rather than among individual persons. He specified a set of standard kinship positions that he called the "effective kinship network".

Traditional analysis (e.g. Murdock, 1971) have argued that societies sometimes proscribe interaction among some kinship positions as an extension of incest taboos. Thus, given this reasoning, kinship positions should be segregated according to the gender of their occupants. White's data provide possibility to test of this hypothesis.

White collected data on the rules governing various kinds of interaction among occupants of his ten standard kinship positions for a sample of 219 societies. For every pair of positions White specified whether or not interaction between their occupants was ever restricted in any society in the sample.

### Source

Freeman, Linton C. (1978) "Segregation in Social Networks" *Sociological Methods and Research* 6(4):411–429

**References**

- Freeman, Linton C. (1978) "Segregation in Social Networks" *Sociological Methods and Research* 6(4):411–429
- Murdock, G. P. (1971) "Cross-Sex Patterns of Kin Behavior" *Ethnology* 1: 359–368
- White, D. R. (1975) "Communicative Avoidance in Social Networks". University of California, Irvine. (mimeo)

**Examples**

```
if( require(igraph, quietly = TRUE) ) {  
  data(Wnet)  
  set.seed(2992)  
  plot(Wnet, layout=layout.fruchterman.reingold,  
        vertex.color=V(Wnet)$gender + 1,  
        vertex.label=V(Wnet)$name, vertex.label.family="",  
        main="White's (1975) data on kinship networks")  
  legend("topleft", col=2:3, legend=c("Woman", "Man"), pch=19)  
}
```

# Index

- \* **datasets**
  - earnings, 6
  - EFnet, 7
  - gender\_roles, 15
  - read\_highland\_tribes, 22
  - Wnet, 27
- \* **graphs**
  - freeman, 10
- \* **segregation measures**
  - assort, 2
  - coleman, 5
  - ei, 7
  - freeman, 10
  - gamix, 14
  - orwg, 20
  - smi, 23
  - ssi, 24

assort, 2, 6, 8, 11, 14, 21, 24, 25

Catania, 4

coauthorship, 5

coleman, 3, 5, 8, 11, 14, 21, 24, 25

earnings, 6

EFnet, 7

ei, 3, 6, 7, 11, 14, 21, 24, 25

fold, 9, 26

freeman, 3, 6, 8, 10, 14, 21, 24, 25

full\_mm, 11

Galesburg2, 13

gamix, 3, 6, 8, 11, 14, 21, 24, 25

gender\_roles, 15

group\_sizes, 15

IBE121, 16

Input\_output, 17

is.directed, 19

isnar, 17

isnar-package (isnar), 17

judge\_net, 18

judge\_net\_bp, 18, 18

lower.tri, 9

mixingm, 3, 19

orwg, 3, 6, 8, 11, 14, 20, 24, 25

pairwise\_redundancy, 21

pr\_invdistance (pairwise\_redundancy), 21

pr\_sharedp (pairwise\_redundancy), 21

read\_highland\_tribes, 22

smi, 3, 6, 8, 11, 14, 21, 23, 25

ssi, 3, 6, 8, 11, 14, 21, 24, 24

symmetrize, 26

upper.tri, 9

Wnet, 27