# Package 'netseg'

October 13, 2022

**Title** Measures of Network Segregation and Homophily

**Version** 1.0-1

**Description** Segregation is a network-level property such that edges between
predefined groups of vertices are relatively less likely. Network homophily
is a individual-level tendency to form relations with people who are similar
on some attribute (e.g. gender, music taste, social status, etc.). In general
homophily leads to segregation, but segregation might arise without
homophily. This package implements descriptive indices measuring
homophily/segregation. It is a computational companion
to Bojanowski & Corten (2014) <doi:10.1016/j.socnet.2014.04.001>.

**Depends** R (>= 2.10)

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

**License** GPL-2

**Encoding** UTF-8

**VignetteBuilder** knitr

**URL** https://mbojan.github.io/netseg/

**BugReports** https://github.com/mbojan/netseg/issues

**LazyData** true

**RoxygenNote** 7.2.1

**Suggests** testthat (>= 2.1.0), knitr, rmarkdown, scales

**NeedsCompilation** no

**Author** Michal Bojanowski [aut, cre] (<https://orcid.org/0000-0001-7503-852X>)

**Maintainer** Michal Bojanowski <michal2992@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-08-25 12:10:06 UTC

# R topics documented:

---

netseg-package                 *netseg: Measures of Network Segregation and Homophily*

---

## Description

Segregation is a network-level property such that edges between predefined groups of vertices are relatively less likely. Network homophily is a individual-level tendency to form relations with people who are similar on some attribute (e.g. gender, music taste, social status, etc.). In general homophily leads to segregation, but segregation might arise without homophily. This package implements descriptive indices measuring homophily/segregation. It is a computational companion to Bojanowski & Corten (2014) doi:10.1016/j.socnet.2014.04.001.

## Author(s)

**Maintainer**: Michal Bojanowski <michal2992@gmail.com> (ORCID)

## References

Bojanowski, Michał, and Rense Corten. 2014. "Measuring Segregation in Social Networks." *Social Networks* 39: 14–32. doi:10.1016/j.socnet.2014.04.001.

## See Also

Useful links:

- Package homepage: https://mbojan.github.io/netseg
- Bug reports: https://github.com/mbojan/netseg/issues

---

assort *Assortativity Coefficient*

---

### Description

Assortativity coefficient is a measure of segregation for social networks due to Newman & Girvan (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 interpreted 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-classyfying 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.

If `object` is of class "igraph" it is required to supply `vattr` with the name of the vertex attribute to calculate intermediate mixing matrix.

For any other classes, `object` is 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(WhiteKinship, "gender")
assort(EF3, "type")

# Values of `assort()` for full networks of different sizes
if( requireNamespace("igraph", quietly = TRUE) ) {
  f <- function(n) {
    gfull <- igraph::make_full_graph(n, directed=FALSE)
    igraph::V(gfull)$type <- rep(1:2, length = igraph::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).

## Usage

```
Catania
```

## Format

Four-by-four numeric matrix with dimnames.

```
#>         female
#> male     black hispanic other white
#>  black      506      32    26    69
#>  hispanic    23     308    38   114
#>  other       10      14    32    47
#>  white       26      46    68   599
```

## 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 )
```

---

Classroom                    *Classroom network*

---

## Description

A directed network of ties "with whom do you like to play with?" collected in a school class of 9-year olds. The data comes from a study undertook by Educational Research Institute (Dolata 2014).

## Usage

```
Classroom
```

## Format

Object of class "igraph" with a directed network of size 26. Vertex attributes:

- gender – with values "Boy" or "Girl".

**References**

Dolata, Roman (ed). (2014). *Czy szkoła ma znaczenie? Zróżnicowanie wyników nauczania po pierwszym etapie edukacyjnym oraz jego pozaszkolne i szkolne uwarunkowania*. Vol. 1. Warsaw: Instytut Badań Edukacyjnych.

---

| coleman | *Coleman's homophily index* |
|---------|------------------------------|

---

**Description**

Colemans's homphily 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**

| | |
|---|---|
| object | R object, see Details for available methods |
| ... | other arguments passed to/from methods |
| gsizes | numeric vector of group sizes |
| loops | logical, whether loops are allowed |
| vattr | 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. Two-dimensional table is interpreted as a contact layer. Three-dimensional table is interpreted as a full mixing matrix $m_{ghy}$ cross-classyfying 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.

If `object` is of class "igraph" it is required to supply `vattr` with the name of the vertex attribute to calculate intermediate mixing matrix.

## 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(requireNamespace("igraph", quietly = TRUE)) {
  # Converting networks to directed
  coleman(igraph::as.directed(WhiteKinship, "mutual"), "gender")
  coleman(igraph::as.directed(EF3, "mutual"), "race")
}
```

---

EF3                         *Example data from Figure III of Echenique and Fryer (2006)*

---

## Description

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

## Usage

```
EF3
```

## Format

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

- name – vertex names of the form "A1" in which letter and number indicate the position in the lattice
- race – values 1 or 2 indicate the two groups

## 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(requireNamespace("igraph", quietly = TRUE)) {
  set.seed(1)
  plot(
    EF3,
    layout = igraph::layout.fruchterman.reingold,
    vertex.color = igraph::V(EF3)$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 interpreted 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-classyfying 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.

If `object` is of class "igraph" it is required to supply `vattr` with the name of the vertex attribute to calculate intermediate mixing matrix.

## Value

Numerical value of the E-I index.

## References

Krackhardt, D., & Stern, R. N. (1988). Informal networks and organizational crises: An experimental simulation. *Social Psychology Quarterly*, 123-140.

## See Also

Other segregation measures: `assort()`, `coleman()`, `freeman()`, `gamix()`, `orwg()`, `smi()`, `ssi()`

## Examples

```
ei(WhiteKinship, "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

| | |
|---|---|
| x | square numeric matrix |
| direction | 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 symetrically 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()](), [symmetrize()]()

**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, 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 |
| loops | logical, whether loops are allowed |
| vattr | character scalar or any vector of length equal to vcount(object), 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.

Originally the index has a discontinuity for network and group size configurations that are characterized by the higher number of between-group ties that is expected under a random graph, for which it returns 0 (as originally described by Freeman (1978)). We removed that truncation such that it returns values betweem -1 and 1.

The original Freeman's formulation involves two groups of vertices. Here it is extended to the arbitrary number of groups. The generalization affects the way in which the expected number of between-group edges under pure random graph is calculated, see Bojanowski & Corten (2014) for details.

The function internally calculates the sizes of groups of vertices in the supplied attribute `vattr`. However, it is possible to override this by specifying "true" type distribution with the `gsizes` argument. It is assumed to be a table (as returned by [table()](#)) or a numeric vector with the group sizes. This may be especially usefull when dealing with large graphs and/or with large number of isolates.

If `object` is a table it is interpreted 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-classyfying 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.

If `object` is of class "igraph" it is required to supply `vattr` with the name of the vertex attribute to calculate intermediate mixing matrix.

Method for mixing matrices

Method for "igraph"s

**Value**

The value of the Freeman's index.

**References**

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

Bojanowski, Michał, and Rense Corten. 2014. "Measuring Segregation in Social Networks." *Social Networks* 39: 14–32. doi:10.1016/j.socnet.2014.04.001

**See Also**

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

**Examples**

```
## White's data from Freeman's article segregation level
freeman(WhiteKinship, "gender")
```

```
# using 'more' argument
freeman(WhiteKinship, "gender")
```

---

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, ...)

## 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 |
| vattr | character, name of vertex attribute |

### Details

The measure varies between `-1/vcount(g)` for dissassortative mixing and 1 for perfect within-group mixing. It takes a value of 0 for proportionate mixing.

If `object` is a table it is interpreted 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-classyfying 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.

If `object` is of class "igraph" it is required to supply `vattr` with the name of the vertex attribute to calculate intermediate mixing matrix.

### 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(WhiteKinship, "gender")
gamix(EF3, "race")
```

---

| group_sizes | *Computing group sizes from square mixing matrices* |
|---|---|

---

## Description

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

---

| mixingm | *Network mixing matrix* |
|---|---|

---

## Description

Creating network mixing matrices (mixingm()) and data frames (mixingdf()).

## Usage

```
mixingm(object, ...)

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

mixingdf(object, ...)

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

## S3 method for class 'igraph'
mixingdf(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 `igraph::is_directed()`. |
| `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 tie sender and tie receiver.

### Value

Function `mixingm()`, 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).

Function `mixingdf()` returns non-zero entries of a mixing matrix (as returned by `mixingm()`), but organized in a data frame with columns:

- `ego`, `alter` – group membership of ego an alter
- `tie` – present only if `full=TRUE`, with `TRUE` or `FALSE` for connected and disconnected dyads respectively
- `n` – counts

### Examples

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

If `object` is a table it is interpreted 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-classyfying 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.

If `object` is of class "igraph" it is required to supply `vattr` with the name of the vertex attribute to calculate intermediate mixing matrix.

## 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(WhiteKinship, "gender")
```

---

| smi | *Segregation Matrix Index* |

---

## Description

Segregation Matrix Index due to Freshtman (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.

If object is a table it is interpreted 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-classyfying 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.

If object is of class "igraph" it is required to supply vattr with the name of the vertex attribute to calculate intermediate mixing matrix.

## 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

Freshtman, 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

```
# smi() needs a directed network
smi( igraph::as.directed(WhiteKinship, "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

| | |
|---|---|
| g | object of class "igraph" representing a network |
| vattr | 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, F., & Fryer Jr, R. G. (2007). A measure of segregation based on social interactions. The Quarterly Journal of Economics, 122(2), 441-485.

### See Also

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

### Examples

```
if(requireNamespace("igraph", quietly = TRUE)) {

### artificial EF data
x <- ssi(EF3, "race")
x


# show it on picture
a <- igraph::V(EF3)$race
# rescale SSI values to use as shades of gray
k <- 1 - scale(x, center=min(x), scale=max(x) - min(x))
plot( EF3, layout=igraph::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(WhiteKinship, "gender")
x

# plot it
a <- match(igraph::V(WhiteKinship)$gender, unique(igraph::V(WhiteKinship)$gender))
k <- 1 - scale(x, center=min(x), scale=max(x) - min(x))
set.seed(1234)
plot( WhiteKinship, layout=igraph::layout.fruchterman.reingold,
main="Node segregation in White's kinship data",
vertex.label.family="",
vertex.label=igraph::V(WhiteKinship)$name,
vertex.color= gray(k),
vertex.shape=c("circle", "csquare")[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 asymetric, 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))
```

---

WhiteKinship                 *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.

### Usage

WhiteKinship

### Format

Object of class "igraph" with an undirected network of size 10. Vertex attribute gender, takes values "male" or "female".

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

# Index