Package ‘assortnet’
February 24, 2023

Type Package

Title Calculate the Assortativity Coefficient of Weighted and Binary Networks

Version 0.20

Date 2023-02-24

Author Damien Farine <damien.farine@anu.edu.au>

Maintainer Damien Farine <damien.farine@anu.edu.au>

Description Functions to calculate the assortment of vertices in social networks. This can be measured on both weighted and binary networks, with discrete or continuous vertex values.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2023-02-24 12:30:06 UTC

R topics documented:

assortnet-package ................................................................. 1
assortment.continuous .......................................................... 2
assortment.discrete ............................................................ 4

Index 6

assortnet-package Calculate the assortativity coefficient of weighted and binary networks
~~ assortnet ~~

Description

Functions to calculate the assortment of vertices in social networks. This can be measured on both weighted and binary networks, with discrete or continuous vertex values.

Details
assortment.continuous

Package: assortnet
Type: Package
Version: 0.20
Date: 2023-02-24
License: GPL2

Author(s)
Maintainer: Damien Farine <damien.farine@anu.edu.au>

References

assortment.continuous  Assortment on continuous vertex values

Description
Calculates the assortativity coefficient for weighted and unweighted graphs with numerical vertex values

Usage
assortment.continuous(graph, vertex_values, weighted = TRUE,
SE = FALSE, M = 1, na.rm = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph</td>
<td>A Adjacency matrix, as an N x N matrix. Can be weighted or binary.</td>
</tr>
<tr>
<td>vertex_values</td>
<td>Values on which to calculate assortment, vector of N numbers</td>
</tr>
<tr>
<td>weighted</td>
<td>Flag: TRUE to use weighted edges, FALSE to turn edges into binary (even if weights are given)</td>
</tr>
<tr>
<td>SE</td>
<td>Calculate standard error using the Jackknife method.</td>
</tr>
<tr>
<td>M</td>
<td>Binning value for Jackknife, where M edges are removed rather than single edges. This helps speed up the estimate for large networks with many edges.</td>
</tr>
<tr>
<td>na.rm</td>
<td>Remove all nodes which have NA as vertex_values from both the network and the vertex_values object. If this is False and NAs are present, an error message will be displayed.</td>
</tr>
</tbody>
</table>
This function returns a named list, with two elements:
Sr the assortativity coefficient $SE$ the standard error

Author(s)
Damien Farine dfarine@orn.mpg.de

References

Examples

```r
# DIRECTED NETWORK EXAMPLE
# Create a random directed network
N <- 20
dyads <- expand.grid(ID1=1:20,ID2=1:20)
dyads <- dyads[which(dyads$ID1 != dyads$ID2),]
weights <- rbeta(nrow(dyads),1,15)
network <- matrix(0, nrow=N, ncol=N)
network[cbind(dyads$ID1,dyads$ID2)] <- weights

# Create random continues trait values
traits <- rnorm(N)

# Test for assortment as binary network
assortment.continuous(network,traits,weighted=FALSE)

# Test for assortment as weighted network
assortment.continuous(network,traits,weighted=TRUE)

# UNDIRECTED NETWORK EXAMPLE
# Create a random undirected network
N <- 20
dyads <- expand.grid(ID1=1:20,ID2=1:20)
dyads <- dyads[which(dyads$ID1 < dyads$ID2),]
weights <- rbeta(nrow(dyads),1,15)
network <- matrix(0, nrow=N, ncol=N)
network[cbind(dyads$ID1,dyads$ID2)] <- weights
network[cbind(dyads$ID2,dyads$ID1)] <- weights

# Create random continues trait values
traits <- rnorm(N)

# Test for assortment as binary network
assortment.continuous(network,traits,weighted=FALSE)
```
asserment.discrete

# Test for assortment as weighted network
asserment.continuous(network, traits, weighted=TRUE)

asserment.discrete

**Assortment on discrete vertex values**

**Description**
Calculates the assortativity coefficient for weighted and unweighted graphs with nominal/categorical vertex values

**Usage**
asserment.discrete(graph, types, weighted = TRUE, SE = FALSE, M = 1, na.rm = FALSE)

**Arguments**
- **graph**: Adjacency matrix, as an N x N matrix. Can be weighted or binary.
- **types**: Values on which to calculate assortment, vector of N labels
- **weighted**: Flag: TRUE to use weighted edges, FALSE to turn edges into binary (even if weights are given)
- **SE**: Calculate standard error using the Jackknife method.
- **M**: Binning value for Jackknife, where M edges are removed rather than single edges. This helps speed up the estimate for large networks with many edges.
- **na.rm**: Remove all nodes which have NA as type from both the network and the types object. If this is False and NAs are present, an error message will be displayed.

**Value**
This function returns a named list, with three elements:
- $r$ the assortativity coefficient
- $SE$ the standard error
- $mixing\_matrix$ the mixing matrix with the distribution of edges or edge weights by category

**Author(s)**
Damien Farine dfarine@orn.mpg.de

**References**
Examples

# DIRECTED NETWORK EXAMPLE
# Create a random directed network
N <- 20
dyads <- expand.grid(ID1=1:20,ID2=1:20)
dyads <- dyads[which(dyads$ID1 != dyads$ID2),]
weights <- rbeta(nrow(dyads),1,15)
network <- matrix(0, nrow=N, ncol=N)
network[cbind(dyads$ID1,dyads$ID2)] <- weights

# Create random discrete trait values
traits <- rpois(N,2)

# Test for assortment as binary network
assortment.discrete(network,traits,weighted=FALSE)

# Test for assortment as weighted network
assortment.discrete(network,traits,weighted=TRUE)

# UNDIRECTED NETWORK EXAMPLE
# Create a random undirected network
N <- 20
dyads <- expand.grid(ID1=1:20,ID2=1:20)
dyads <- dyads[which(dyads$ID1 < dyads$ID2),]
weights <- rbeta(nrow(dyads),1,15)
network <- matrix(0, nrow=N, ncol=N)
network[cbind(dyads$ID1,dyads$ID2)] <- weights
network[cbind(dyads$ID2,dyads$ID1)] <- weights

# Create random discrete trait values
traits <- rpois(N,2)

# Test for assortment as binary network
assortment.discrete(network,traits,weighted=FALSE)

# Test for assortment as weighted network
assortment.discrete(network,traits,weighted=TRUE)
Index

* package
  assortnet-package, 1

assortment.continuous, 2
assortment.discrete, 4
assortnet (assortnet-package), 1
assortnet-package, 1