

Package ‘Neighboot’

September 21, 2020

Title Neighborhood Bootstrap Method for RDS

Version 1.0.0

Date 2020-09-09

Author Mamadou Yauck [aut, cre], Erica E. M. Moodie [aut]

Maintainer Mamadou Yauck <mamadou.yauck@mcgill.ca>

Description A bootstrap method for Respondent-Driven Sampling (RDS) that relies on the underlying structure of the RDS network to estimate uncertainty.

License GPL-3

LazyData true

RoxygenNote 7.1.1

Imports magrittr, RDStreeboot, igraph, RDS, dplyr

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-09-21 08:30:07 UTC

R topics documented:

Neighboot	2
pop.network	3
to.rds	4
Index	5

 Neighboot

Compute standard errors and confidence intervals

Description

This function estimate standard errors and compute confidence intervals from an RDS sample using the neighborhood bootstrap method.

Usage

```
Neighboot(RDS.data, quant=c(0.025, 0.975),
          method=c("percentile", "Wald"), B=1000)
```

Arguments

RDS.data	A list containing the following objects: nodes a numeric vector containing IDs edges a list containing two vectors: node1 for the recruiter's ID and node2 for the recruit's ID. traits a data frame containing respondents' traits. degree a vector containing each node's degree, or number of social connections.
quant	a vector of positive integers between 0 and 1, representing quantiles to be estimated.
method	a character string representing the method for computing confidence intervals, either percentile or Wald. Default is percentile.
B	the number of bootstrap repetitions. Default is 1000.

Details

The function Neighboot compute standard errors and confidence intervals using the neighborhood bootstrap method for RDS. Confidence intervals can be computed using the percentile method or the studentized method.

Value

A matrix of estimated standard errors and quantiles. Each row represents a trait.

Author(s)

Mamadou Yauck <mamadou.yauck@mcgill.ca> and Erica E. M. Moodie.

Examples

```
#Load the synthetic population network dataset.
data("pop.network")

#Draw an RDS sample from the simulated network using the sampleRDS function
#from the package RDStreeboot.
require(RDStreeboot)
RDS.samp <- sample.RDS(pop.network$traits, pop.network$adj.mat, 200, 10,
  3, c(1/6,1/3,1/3,1/6), FALSE)

#Compute 95\% confidence intervals using the percentile method
Neighboot(RDS.data=RDS.samp, quant=c(0.025, 0.975),method="percentile", B=100)
```

pop.network	<i>Population network dataset</i>
-------------	-----------------------------------

Description

A synthetic population network.

Usage

```
pop.network
```

Format

A list containing two objects:

`traits` a data frame of 4 columns containing the nodes' IDs in its first column.

`adj.mat` the adjacency matrix of the population network.

Details

The population network has 2000 nodes and has a density of 5%. Three binary attributes are measured on each node: age (less than 30=1), gender (male=1) and education (diploma lower than college's=1).

Examples

```
#load the population network dataset
data("pop.network")
```

to.rds *Transform an sample.RDS object to an rds.data.frame object.*

Description

This function transforms an output from the `sample.RDS` function of the **RDSreeboot** package to an `rds.data.frame` object of the **RDS** package.

Usage

```
to.rds(RDS.data)
```

Arguments

`RDS.data` A list containing the following objects:

- `nodes` a numeric vector containing IDs
- `edges` a list containing two vectors: `node1` for the recruiter's ID and `node2` for the recruit's ID.
- `traits` a data frame containing respondents' traits.
- `degree` a vector containing each node's degree, or number of social connections.

Value

An `rds.data.frame` object.

Author(s)

Mamadou Yauck <mamadou.yauck@mcgill.ca> and Erica E. M. Moodie.

Examples

```
#Load the synthetic population network dataset.
data("pop.network")

#Draw an RDS sample from the simulated network using the samplerDS function
#from the package RDSreeboot.
require(RDSreeboot)
RDS.samp <- sample.RDS(pop.network$traits, pop.network$adj.mat, 200, 10,
  3, c(1/6,1/3,1/3,1/6), FALSE)

#Tranform RDS.samp to an rds.data.frame object
require(RDS)
to.rds(RDS.data=RDS.samp)
```

Index

* **datasets**

pop.network, 3

Neighboot, 2

pop.network, 3

to.rds, 4