Package 'chords'

April 20, 2016

Type Package

Title Estimation in respondent driven samples.

Version 0.92.5

Date 2016-04-20

Author Jonathan Rosenblatt

Maintainer Jonathan Rosenblatt <john.ros@gmail.com>

Description Maximum likelihood estimation in respondent driven samples.

License GPL-2

Imports MASS

Suggests Matrix

NeedsCompilation no

R topics documented:

chords-package 1
brazil 2
estimate.b.k 3
getTheta 5
initializeRdsObject 6
makeRdsSample 8
thetaSmoothingNks 9

Index **10**

`chords-package` *Population size estimation for respondent driven sampling*

Description

Estimates population size and degree distribution in respondent driven samples (RDS).

Details

Maximum likelihood estimation of population size using the methodology in the reference. The fundamental idea is of modeling the sampling as an epidemic process. See `estimate.b.k` for details.

Author(s)

Jonathan D. Rosenblatt <john.ros.work@gmail.com>

References

[1] Berchenko, Yakir, Jonathan Rosenblatt, and Simon D. W. Frost. "Modeling and Analyzing Respondent Driven Sampling as a Counting Process."" arXiv:1304.3505, April 11, 2013. HTTP://arXiv.org/abs/1304.3505.

`brazil` *Heavy Drug Users in Curitiba*

Description

A respondent driven sample of heavy drug users in Curitiba.

Usage

```
data("brazil")
```

Format

A data frame with 303 observations on the following 8 variables.

`MyUniID` Subject's ID.

`NS1` Subject's self reported degree.

`refCoupNum` Reference coupon no.

`coup1` Supplied coupon.

`coup2` Supplied coupon.

`coup3` Supplied coupon.

`interviewDt` Time of interview. See details.

`interviewDt2` Deprecated.

Details

The format of the data is essentially that of the RDS file format as specified in page 7 in the RDS Analysis tool manual: http://www.respondentdrivensampling.org/reports/RDSAT_7.1-Manual_2012-11-25.pdf.

The RDS format has been augmented with the time of interview (`interviewDt` variable) required for the methodology in [1].

The `interviewDt` variable encodes the time of interview. For the purpose of calling `estimate.b.k` the scale and origin are imaterial. We thus use an arbitrary efficient encoding which might not adhere to the original scale.

For full details see the Source section.

Source

[1] Salganik, M.J., Fazito, D., Bertoni, N., Abdo, A.H., Mello, M.B., and Bastos, F.I. (2011). "Assessing Network Scale-up Estimates for Groups Most at Risk of HIV/AIDS: Evidence From a Multiple-Method Study of Heavy Drug Users in Curitiba, Brazil." American Journal of Epidemiology, 174(10): 1190-1196. http://dx.doi.org/10.1093/aje/kwr246

And http://opr.princeton.edu/archive/nsum/

References

[2] Berchenko, Yakir, Jonathan Rosenblatt, and Simon D. W. Frost. "Modeling and Analyzing Respondent Driven Sampling as a Counting Process."" arXiv:1304.3505, April 11, 2013. HTTP://arXiv.org/abs/1304.3505.

Examples

```
# see estimate.b.k()
```

`estimate.b.k` *Estimate population size from RDS sample*

Description

Performs maximum likelihood estimation of a population size from a respondent driven sample. The underlying idea is that the sample spreads like an epidemic in the target population as described in the reference.

Usage

```
estimate.b.k(rds.object,
const=1,
impute.Nks=TRUE)
```

Arguments

`rds.object`	A object of class `rds-object` as constructed by `initializeRdsObject`
`const`	A scaler value for numerical stability. Should not be altered.
`impute.Nks`	Should inestimable values be imputed?

Details

See the reference for a description of the likelihood problem solved. Optimization is performed by noting that likelihood is coordinate-wise convex, thus amounts to a series of line-searches.

If `impute.Nk` is enabled (default), a simpe heuristic is used to estimate degree frequencies which are inestimable using maximum likelihood. This essentially consists of scaling the observed degree by the average observed-to-estimated factor over all estimable frequencies.

Value

A list with the following components:

`call`	The function call.
`Nk.estimates`	The estimated degree frequencies.
`log.bk.estimates`	The estimated sampling rates for each degree.

Note

This function replaces the `estimate.rds` function from versions prior to 0.8 which is now deprecated. Differences between the functions include: 1- Likelihood is now computed in continuous time. The sample data is thus assumed to have an interviewDt column. 2- Optimization now exploits the coordinate-wise convexity of the likelihood function for efficient solving.

Author(s)

Jonathan D. Rosenblatt <john.ros.work@gmail.com>

References

[1] Berchenko, Yakir, Jonathan Rosenblatt, and Simon D. W. Frost. "Modeling and Analyzing Respondent Driven Sampling as a Counting Process."" arXiv:1304.3505, April 11, 2013. HTTP://arXiv.org/abs/1304.3505.

See Also

`initializeRdsObject`, `makeRdsSample`, `getTheta`

Examples

```
# Import data in RDS file format:
data(brazil)
# Initialize RDS object:
rds.object2<- initializeRdsObject(brazil)
# Estimate:
rds.object2$estimates <- estimate.b.k(rds.object = rds.object2 )

# View estimates:
plot(rds.object2$estimates$Nk.estimates, type='h')
# Population size estimate:
sum(rds.object2$estimates$Nk.estimates)
plot(rds.object2$estimates$log.bk.estimates, type='h')


## Recover theta assuming b.k=b_0*k^theta
getTheta(rds.object2)

# How many degrees were imputed?:
table(rds.object2$estimates$convergence)
```

`getTheta` *Recover the "discoverability coefficient".*

Description

Estimates the effect of the degree on the rate of sampling. Also known as the "coefficient of discoverability" in the oil-discovery literature [2]. Formally, we estimate $theta$ and $beta_0$ assuming that $beta_k := beta_0 * k^t heta$.

Usage

```
getTheta(rds.object, bin=1, robust=TRUE)
```

Arguments

`rds.object`	A `rds-object` with a `estimates` component as returned by `estimate.b.k`
`bin`	Bin degree counts. See Note.
`robust`	Should $beta_0$ and $theta$ be recovered from $beta_k$ using a robust method (default) or not.

Value

A list including the following components:

`log.beta_0`	The log of $beta_0$ in $beta_k := beta_0 * k^t heta$.
`theta`	$theta$ in $beta_k := beta_0 * k^t heta$.

Note

If degree counts have been binned by `initializeRdsObject` (for variance reduction), the same value has to be supplied to `getTheta` for correct estimation.

Author(s)

Jonathan D. Rosenblatt <john.ros.work@gmail.com>

References

[1] Berchenko, Yakir, Jonathan Rosenblatt, and Simon D. W. Frost. "Modeling and Analyzing Respondent Driven Sampling as a Counting Process."" arXiv:1304.3505. HTTP://arXiv.org/abs/1304.3505.
[2] Bloomfield, P., K.S. Deffeyes, B. Silverman, G.S. Watson, Y. Benjamini, and R.A. Stine. Volume and Area of Oil Fields and Their Impact on Order of Discovery, 1980. http://www.osti.gov/scitech/servlets/purl/6037591

See Also

`estimate.b.k`, `initializeRdsObject`, `makeRdsSample`

Examples

```
# See in estimate.b.k()
```

`initializeRdsObject` *Construct a* `rds-object` *from a data.frame.*

Description

Given a data frame with the appropriate variables, initializes a `rds-object` with the components required by the `estimate.b.k` function for estimation.

Usage

```
initializeRdsObject(rds.sample, bin=1L, seeds=1L)
```

Arguments

`rds.sample`	A data frame with required columns. See Details.
`bin`	The number of degrees fo bin together. See details.
`seeds`	The number of seed recruiters. See details.

Details

The essence of the function is in recovering the sampling snowball required by `estimate.b.k`. The function allows for recruiters to enter and exit the sampling snowball. The number of seed recruiters is typically not specified in an RDS file. The `seeds` argument is a workaround that allows to specify directly this number.

The `rds.sample` object is assumed to be a data frame with the following column names:

1. MyUniIDan identifier of the sampling unit.[not required]
2. NS1The reported degree.[required]
3. refCoupNum The number of the referring coupon.
4. coup1The number of the 1st supplied coupon. NA if none. [required].
5. coupXThe number of the Xth supplied coupon. NA if none.[not required]
6. interviewDtThe time of the interview. In numeric representation from some origin. Ties are not allowed.

See `brazil` for a sample data.frame.

If the sample is short, stabilization of degree estimates can be achieved by binning degrees together. This can be done with the `bin` argument. Note however that the interpretation of the estimated degree counts is now different as the k'th degree is actually the k'th bin, which is only proportional to k. An exception is the function `getTheta` which also accepts a `bin` argument for proper estimation of $theta$.

Value

A list with the following components.

- rds.sampleThe input data frame. After ordering along time of arrival.
- I.tThe sampling snowball. A list including the following items: I.tAn integer of the count of the sampling individuals at the moments of recruitment. degree.inAn integer with the degree of an added recruiter at the moments of recruitment. degree.outAn integer with the degree of a removed recruiter at the moment of recruitment.
- original.orderingThe order of the arrivals as was inputed in `rds.sample$interviewDt`
- estimatesA placeholder for the future output of `estimate.b.k`

Author(s)

Jonathan D. Rosenblatt <john.ros.work@gmail.com>

References

[1] Berchenko, Yakir, Jonathan Rosenblatt, and Simon D. W. Frost. "Modeling and Analyzing Respondent Driven Sampling as a Counting Process."" arXiv:1304.3505, April 11, 2013. HTTP://arXiv.org/abs/1304.3505.

See Also

`estimate.b.k`, `makeRdsSample`, `brazil`

Examples

```
# See examples in estimate.b.k()
```

`makeRdsSample` *Generate a synthetic (simulated) RDS sample.*

Description

Generates a sample from the sampling process assumed in the reference. Well, actually, only the sufficient statistics required by `estimate.b.k` are returned.

Usage

```
makeRdsSample(N.k, b.k, sample.length)
```

Arguments

`N.k`	An integer vector with the population frequency of each degree.
`b.k`	A numeric vector of the sampling rates of each degree.
`sample.length`	The length of the sample. Specified as the number of recruitees before termination.

Value

An object of class `rds-object` suitable for applying `estimate.b.k`.

Note

The simulator does not prodice a whole RDS sample, but rather the sufficient statistics required for applying `estimate.b.k`.

Author(s)

Jonathan D. Rosenblatt <john.ros.work@gmail.com>

References

[1] Berchenko, Yakir, Jonathan Rosenblatt, and Simon D. W. Frost. "Modeling and Analyzing Respondent Driven Sampling as a Counting Process."" arXiv:1304.3505, April 11, 2013. HTTP://arXiv.org/abs/1304.3505.

See Also

`estimate.b.k`

Examples

```
# Generate data:
true.Nks <- rep(0,100); true.Nks[c(2,100)] <- 1000
theta <- 1e-1
true.log.bks <- rep(-Inf, 100);true.log.bks[c(2,100)] <- theta*log(c(2,100))
sample.length <- 1000L
rds.simulated.object <- makeRdsSample(
  N.k =true.Nks ,
  b.k = exp(true.log.bks),
  sample.length = sample.length)

# Estimate:
rds.simulated.object$estimates <- estimate.b.k(rds.object = rds.simulated.object )
chords:::compareNkEstimate(rds.simulated.object$estimates$Nk.estimates, true.Nks)
```

`thetaSmoothingNks` *Smooth estimates degree frequencies.*

Description

Smoothes estimated N_k by assuming that $beta_k = beta_0 * k^t heta$.

Usage

```
thetaSmoothingNks(rds.object,...)
```

Arguments

`rds.object`	A `rds-object` class object as returned by `estimate.b.k`
`...`	Firther arguments passed to `getTheta`.

Value

A numeric vector of smoothed N_k values.

Author(s)

Jonathan D. Rosenblatt <john.ros.work@gmail.com>

See Also

`estimate.b.k`; `getTheta`

Examples

```
# See estimate.b.k()
```

Index

∗Topic **datasets**
brazil, 2

brazil, 2, *7*

chords *(*chords-package*)*, 1
chords-package, 1

estimate.b.k, *2, 3*, 3, *5–9*

getTheta, *4*, 5, *7*, *9*

initializeRdsObject, *4*, *6*, 6

makeRdsSample, *4*, *6, 7*, 8

thetaSmoothingNks, 9