Package ‘sps’

February 9, 2022

Title Sequential Poisson Sampling

Version 0.1.3

Description

Sequential Poisson sampling is a method for drawing probability-proportional-to-size samples with a given number of units, and is commonly used for price-index surveys. This package gives functions to draw stratified sequential Poisson samples according to the method by Ohlsson (1998, ISSN:0282-423X), and generate bootstrap replicate weights according to the generalized bootstrap method by Beaumont and Patak (2012, <doi:10.1111/j.1751-5823.2011.00166.x>).

Depends R (>= 3.5)

Imports stats

License MIT + file LICENSE

Encoding UTF-8

URL `https://github.com/marberts/sps`

NeedsCompilation no

Author Steve Martin [aut, cre, cph] (<`https://orcid.org/0000-0003-2544-9480`>),
Justin Francis [ctb]

Maintainer Steve Martin <`stevemartin041@gmail.com`>

Repository CRAN

Date/Publication 2022-02-09 15:30:02 UTC

R topics documented:

`sps-package` *Sequential Poisson Sampling*

Description

Sequential Poisson sampling is a method for drawing probability-proportional-to-size samples with a given number of units, and is commonly used for price-index surveys. This package gives functions to draw stratified sequential Poisson samples according to the method by Ohlsson (1998), and generate bootstrap replicate weights according to the generalized bootstrap method by Beaumont and Patak (2012).

Usage

Given a vector of sizes for units in a population (e.g., revenue for sampling businesses) and a desired sample size, a stratified sequential Poisson sample can be drawn with the `sps()` function. The design weights for a sample can then be used to generate bootstrap replicate weights with the `sps_repweights()` function.

Sequential Poisson sampling is often used to sample data for price indexes. Balk (2008, chapter 5) discusses the construction of price indexes when data are sampled using probability-proportional-to-size methods, and their resulting statistical properties (although these results are not directly applicable to sequential Poisson sampling because the method is only approximately proportional to size; see Ohlsson (1998, proposition 2.2) for details). The CPI manual (2020, chapter 4) details other sampling methods that are used for price data.

Author(s)

Maintainer: Steve Martin <`stevemartin041@gmail.com`>

Other contributors:

- Justin Francis

References

Balk, B. M. (2008). *Price and Quantity Index Numbers*. Cambridge University Press.

Beaumont, J.-F. and Patak, Z. (2012). On the Generalized Bootstrap for Sample Surveys with Special Attention to Poisson Sampling. *International Statistical Review*, 80(1): 127-148.

ILO, IMF, OECD, Eurostat, UN, and World Bank. (2020). *Consumer Price Index Manual: Theory and Practice*. International Monetary Fund.

Ohlsson, E. (1998). Sequential Poisson Sampling. *Journal of Official Statistics*, 14(2): 149-162.

See Also

`https://github.com/marberts/sps`

`sps` *Stratified sequential Poisson sampling*

Description

Draw a stratified probability-proportional-to-size sample according to the sequential Poisson sampling method by Ohlsson (1998), with the option of an allocation proportional to size.

Usage

```
sps(x, n, s = rep(1, length(x)), prn = NULL)

## S3 method for class 'sps'
weights(object, ...)

prop_allocation(x, N, s = rep(1, length(x)))
```

Arguments

`x` A strictly positive and finite numeric vector of sizes for units in the population (e.g., revenue for drawing a sample of businesses).

`n` A positive vector of integers giving the sample sizes for each stratum, ordered according to the levels of `s`. Non-integers are truncated towards 0.

`s` A factor, or something that can be coerced into one, giving the strata associated with `x`. The default is to place all units into a single stratum.

`prn` A numeric vector of permanent random numbers distributed uniform between 0 and 1, the same length as `x`. The default does not use permanent random numbers.

`object` An object of class `sps`, as made by `sps()`.

`N` A positive integer giving the total sample size across all strata (i.e., `sum(n)`). Non-integers are truncated towards 0.

`...` Further arguments passed to or used by methods.

Details

The details of the sequential Poisson sampling procedure are in sections 2.2 to 2.4 of Ohlsson (1998). Briefly, for a single stratum, all units in the population with an inclusion probability, $nx/\sum x$, greater than or equal to 1 are placed into a take-all stratum. This process is repeated until all the inclusion probabilities are less than 1.

The remaining units in the sample belong to the take-some stratum, and are drawn by assigning each unit a value $\xi = u/x$, where u is from $U(0, 1)$. If `prn != NULL`, then the corresponding values for `prn` are used instead of generating random values. The units with the smallest values for ξ are included in the sample. In the unlikely event of a tie, the first unit is included in the sample. This is the same method used by `PROC SURVEYSELECT` in SAS with `METHOD = SEQ_POISSON`.

The prop_allocation() function gives a sample size for each stratum that is proportional to x using largest-remainder rounding (so that they add up to N). In cases where the number of units in a stratum is smaller than its allocation, the allocation for that stratum is set to the number of available units in that stratum, with the remaining sample size reallocated to other strata proportional to x. This is similar to PROC SURVEYSELECT in SAS with ALLOC = PROPORTIONAL.

Value

sps() returns an object of class sps. This is a numeric vector of indices for the units in the population that form the sample, along with a weights attribute that gives the design weights for each unit in the sample (keeping in mind that sequential Poisson sampling is only approximately probability-proportional-to-size), and a levels attribute that gives whether a sampled unit belongs to the take-all stratum or take-some stratum. weights() can be used to access the design weights attribute of an sps object, and levels() can be used to access the strata. Mathematical and binary/unary operators strip these attributes, as does replacement.

prop_allocation() returns a named numeric vector of sample sizes for each stratum in s.

References

Ohlsson, E. (1998). Sequential Poisson Sampling. *Journal of Official Statistics*, 14(2): 149-162.

See Also

sps_repweights for generating bootstrap replicate weights.

The UPpoisson() function in the **sampling** package for regular Poisson sampling.

The **pps** package for other probability-proportional-to-size sampling methods.

The strAlloc() function in the **PracTools** package for other allocation methods.

Examples

```
x <- c(1:10, 100) # sizes in the population

# Draw a sample
(samp <- sps(x, 5))

# Get the design (inverse probability) weights
weights(samp)

# All units except 11 are in the take-some (TS) stratum
levels(samp)

# Example of a stratified sample
strata <- rep(letters[1:4], 5)
sps(1:20, c(4, 3, 3, 2), strata)

# Proportional allocation
(allocation <- prop_allocation(1:20, 12, strata))
sps(1:20, allocation, strata)

# It can be useful to set 'prn' in order to extend the sample
# to get a fixed net sample
u <- runif(11)
(samp <- sps(x, 6, prn = u))
sps(x[-samp[5]], 5, prn = u[-samp[5]]) # removing unit 5 gives the same net sample
```

`sps_repweights` *Bootstrap replicate weights for sequential Poisson sampling*

Description

Produce bootstrap replicate weights that are appropriate for sequential Poisson sampling using the generalized bootstrap method by Beaumont and Patak (2012).

Usage

```
sps_repweights(w, B = 1000, tau = 1, dist = NULL)
```

Arguments

- `w` A numeric vector of design weights for a sequential Poisson sample.
- `B` An integer that gives the number of bootstrap replicates (1,000 by default). Non-integers are truncated towards 0.
- `tau` A number greater than or equal to 1 that gives the rescale factor for the bootstrap weights. Setting to 1 (the default) does not rescale the weights.
- `dist` A function that produces random deviates with mean 0 and standard deviation 1, such as `rnorm`. The default uses the pseudo-population method from section 4.1 of Beaumont and Patak (2012).

Details

The details of the method for generating bootstrap replicates are in sections 4 and 6 of Beaumont and Patak (2012). Briefly, their method involves finding a vector of adjustments a for each bootstrap replicate and calculating the replicate weights as aw, where w is a vector of design weights (inverse of the inclusion probabilities).

The default pseudo-population method randomly rounds w for each replicate to produce a collection of integer weights w' that are used to generate a random vector b from a binomial distribution. The vector of adjustments is then $a = 1 + b - w'/w$. Specifying a deviates-generating function for `dist` will use this function to make a random vector d that is used to make an adjustment $a = 1 + d\sqrt{1 - 1/w}$.

These adjustments can be rescaled by a value $\tau \geq 1$ to prevent negative replicate weights. With this rescaling, the adjustment becomes $(a + \tau - 1)/\tau$. If $\tau > 1$ then the resulting bootstrap variance estimator should be multiplied by τ^2.

Value

An object of class sps_brw. This is a matrix of bootstrap replicate weights with B columns (one for each replicate) and length(w) rows (one for each unit in the sample), with the value of tau as an attribute. Any names for w are kept as row names.

There are currently no interesting methods for this class.

References

Beaumont, J.-F. and Patak, Z. (2012). On the Generalized Bootstrap for Sample Surveys with Special Attention to Poisson Sampling. *International Statistical Review*, 80(1): 127-148.

See Also

sps for drawing a sequential Poisson sample.

Examples

```
x <- c(1:10, 100) # sizes in the population

# Draw a sample
(samp <- sps(x, 5))

# Make some bootstrap replicates
sps_repweights(weights(samp), B = 5) # pseudo-population method

sps_repweights(weights(samp), B = 5, dist = rnorm) # standard-normal method

sps_repweights(weights(samp), B = 5, dist = function(x) rexp(x) - 1) # exponential method
```

Index