Package 'sps'

November 23, 2022

Title Sequential Poisson Sampling

Version 0.3.0

Description Sequential Poisson sampling is a variation of Poisson sampling 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 appropriate 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

BugReports https://github.com/marberts/sps/issues

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 2022-11-23 07:30:02 UTC

R topics documented:

sps-package *Sequential Poisson Sampling*

Description

Sequential Poisson sampling is a variation of Poisson sampling 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 appropriate 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. Allocations are often proportional to size when drawing such samples, and the `prop_allocation()` function provides a variety of methods for generating proportional-to-size allocations. Once the sample is drawn, the design weights for the 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. The CPI manual (2020, chapter 4) describes other methods for sampling 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 using the (sequential) Poisson method, with the option of an allocation proportional to size.

Usage

```
## Sequential Poisson sampling
sps(x, n, strata = gl(1, length(x)), prn = runif(length(x)))

## Ordinary Poisson sampling
ps(x, n, strata = gl(1, length(x)), prn = runif(length(x)))

inclusion_prob(x, n, strata = gl(1, length(x)))

## Allocations
prop_allocation(
  x, N,
  strata = gl(1, length(x)),
  initial = 0,
  divisor = function(a) a + 1
)

expected_coverage(x, N, strata = gl(1, length(x)))

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

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 size for each stratum, ordered according to the levels of strata. Non-integers are truncated towards 0.
strata	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, instead generating a random vector when the function is called.
N	A positive integer giving the total sample size across all strata (i.e., sum(n)). Non-integers are truncated towards 0.

`initial`	A positive vector of integers giving the initial (or minimal) allocation for each stratum, ordered according to the levels of `strata`. A single integer is recycled for each stratum using a special algorithm to ensure a feasible allocation; see details. Non-integers are truncated towards 0. The default allows for strata with no units.
`divisor`	A divisor function for the divisor (highest-averages) apportionment method. The default uses the Jefferson (D'Hondt) method. See details for other possible functions.
`object`	An object of class `sps`, as made by `sps()` or `ps()`.
`...`	Further arguments passed to or used by methods.

Details

Sampling: The `sps()` function draws a sample according to the sequential Poisson procedure, the details of which are given by Ohlsson (1998, section 2). 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. The inclusion probabilities for the remaining take-some units are then recalculated, and this process repeats until all the inclusion probabilities are less than or equal to 1. The `inclusion_prob()` function computes these stratum-wise inclusion probabilities.

A sample of take-some units is drawn by assigning each unit a value $\xi = u/x$, where u is a random deviate from the uniform distribution between 0 and 1. The units with the smallest values for ξ are included in the sample, along with the take-all units, resulting in a fixed sample size at the expense of the sampling procedure being only approximately probability-proportional-to-size. 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`.

Ordinary Poisson sampling follows the same procedure as above, except that all units with $\xi < n/\sum x$ are included in the sample; consequently, while it does not contain a fixed number of units, the procedure is strictly probability-proportional-to-size. Despite this difference, the standard Horvitz-Thompson estimator for the total is asymptotically unbiased, normally distributed, and equally efficient under both procedures. It is in this sense that sequential Poisson sampling is approximately probability-proportional-to-size. The `ps()` function draws a sample using the ordinary Poisson method.

Allocations: The `prop_allocation()` function gives a sample size for each stratum that is proportional to the sum of `x` across strata and adds up to `N`. This is done using the divisor (highest-averages) apportionment method (Balinksi and Young, 1982, Appendix A), for which there are a number of different divisor functions:

Jefferson/D'Hondt	`function(a) a + 1`
Webster/Sainte-Laguë	`function(a) a + 0.5`
Imperiali	`function(a) a + 2`
Huntington–Hill	`function(a) sqrt(a * (a + 1))`
Danish	`function(a) a + 1 / 3`
Adams	`function(a) a`
Dean	`function(a) a * (a + 1) / (a + 0.5)`

Note that a divisor function with $d(0) = 0$ (i.e., Huntington-Hill, Adams, Dean) requires an initial allocation of at least 1 for all strata. In all cases, ties are broken according to the levels of `strata`;

reordering the levels of `strata` can therefore result in a different allocation.

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, with the remaining sample size reallocated to other strata proportional to `x`. This is similar to `PROC SURVEYSELECT` in SAS with `ALLOC = PROPORTIONAL`.

Passing a single integer for the initial allocation first checks that recycling this value for each stratum does not result in an allocation larger than the sample size. If it does, then the value is reduced so that recycling does not exceed the sample size. This recycled vector can be further reduced in cases where it exceeds the number of units in a stratum, the result of which is the initial allocation. This special recycling ensures that the initial allocation is feasible.

The `expected_coverage()` function gives the average number of strata covered by ordinary Poisson sampling without stratification. As sequential and ordinary Poisson sampling have the same sample size on average, this gives an approximation for the coverage under sequential Poisson sampling. This function can also be used to calculate, e.g., the expected number of enterprises covered within a stratum when sampling business establishments.

Value

`sps()` and `ps()` return 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 (inverse probability) 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.

`inclusion_prob()` returns a numeric vector of inclusion probabilities for each unit in the population.

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

`expected_coverage()` returns the expected number of strata covered by the sample design.

References

Balinksi, M. L. and Young, H. P. (1982). *Fair Representation: Meeting the Ideal of One Man, One Vote*. Yale University Press.

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

See Also

`sps_repweights` for generating bootstrap replicate weights.

`UPpoisson` and `inclusionprobabilities` in the **sampling** package for ordinary Poisson sampling and calculating inclusion probabilities. They are largely the same as `ps` and `inclusion_prob`, but for a single stratum.

`strAlloc` in the **PracTools** package for other allocation methods.

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

Examples

```
# Make a population with units of different size
x <- c(1:10, 100)

# Draw a sequential Poisson 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)

# Ordinary Poisson sampling gives a random sample size for the
# take-some stratum
ps(x, 5)

# Example of a stratified sample
strata <- rep(letters[1:4], each = 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))

# Removing unit 5 gives the same net sample
sps(x[-samp[5]], 6, prn = u[-samp[5]])
```

---

sps_repweights *Bootstrap replicate weights for sequential Poisson sampling*

Description

Produce bootstrap replicate weights that are appropriate for Poisson sampling, and therefore approximately correct for sequential Poisson sampling.

Usage

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

Arguments

w	A numeric vector of design (inverse probability) 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 presented by Beaumont and Patak (2012, sections 4 and 6). 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 the 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.

As an alternative to the bootstrap, Ohlsson (1998, equations 2.12 and 2.13) proposes an analytic estimator for the variance of the total $\hat{Y} = \sum wy$ under sequential Poisson sampling:

$$V(\hat{Y}) = \frac{n}{n-1} \sum \left(1 - \frac{1}{w}\right) \left(wy - \frac{\hat{Y}}{n}\right)^2 .$$

Replacing the left-most correction by $n/(m-1)$, where m is the number of units in the sample, gives a similar estimator for ordinary Poisson sampling.

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.

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

See Also

sps for drawing a sequential Poisson sample.

bootstrapFP (with method = "wGeneralised") in the **bootstrapFP** package for calculating the variance of Horvitz-Thompson estimators using the generalized bootstrap.

Examples

```
# Make a population with units of different size
x <- c(1:10, 100)

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

# Make some bootstrap replicates
dist <- list(
  pseudo_population = NULL,
  standard_normal = rnorm,
  exponential = function(x) rexp(x) - 1,
  uniform = function(x) runif(x, -sqrt(3), sqrt(3))
)

lapply(dist, sps_repweights, w = weights(samp), B = 5, tau = 2)
```

Index