Package ‘surveyweights’

December 10, 2019

Type Package

Title Calculate weights from sampling frames

Version 0.2.0

Author Martin Barner

Maintainer Martin Barner <martin.barner@impact-initiatives.org>

Description Calculate survey weights from sampling frames and data ``on the fly'', and allows you to combine weights of multiple sampling frames.

License GPL-3

Encoding UTF-8

LazyData true

Depends magrittr

Roxygen list(markdown = TRUE)

RoxygenNote 6.0.1

Suggests knitr,
rmarkdown,
testthat

VignetteBuilder knitr

R topics documented:

`combine_weighting_functions` *Combine weight functions from two sampling frames*

Description

With multi-stage sampling, it is sometimes necessary to combine the weights from two sampling frames (e.g. stratified cluster sampling). This function let's you create a new weight function from two existing weight functions created with weighting_fun_from_samplingframe().

Usage

```
combine_weighting_functions(weights_function_1, weights_function_2)
```

Arguments

`weights_function_1`
The weight function from the *outer* sampling frame (the 'larger' scale; Records in one group of the *outer* sampling frame can belong to different strata in the *inner* sampling frame.)

`weights_function_2`
The weight function from the *inner* sampling frame (the 'smaller' scale; Records in the same group of the *inner* sampling frame must also belong to the same group in the *outer* sampling frame.)

Details

The returned function combines two sets of weights so that

- the sum of weights of each sampling frame's groups remain proportional to each other
- the total sum of weights equals the number of rows in the input data frame

Value

returns a function that takes a dataframe as input and returns a vector of weights

`hasdata` *remove different kinds of missing values from a vector*

Description

namely NULL, NA, "" and any non-finite values

Usage

```
hasdata(x, return.index = F)
```

`stratify.count.sample` *Get sample sizes for weight calculations. Checks if all strata appear in sampling frame*

Description

Get sample sizes for weight calculations. Checks if all strata appear in sampling frame

Usage

```
stratify.count.sample(data.strata, sf.strata)
```

`stratify.weights` *estimates sampling probabilities from sample size and strata populations.*

Description

estimates sampling probabilities from sample size and strata populations.

Usage

```
stratify.weights(pop_strata, sample_strata)
```

`weighting_fun_from_samplingframe` *creates a weighting function from a sampling frame*

Description

creates a weighting function from a sampling frame

Usage

```
weighting_fun_from_samplingframe(sampling.frame, data.stratum.column,
  sampling.frame.population.column = "population",
  sampling.frame.stratum.column = "stratum", data = NULL)
```

Arguments

`data.stratum.column`
: data column name that holds the record's strata names

`sampling.frame.population.column`
: sampling frame name of column holding population counts. defaults to "population"

`sampling.frame.stratum.column`
: sampling frame name of column holding stratum names. defaults to "stratum". Stratum names must match exactly values in:

`data`
: optional but recommended: you can provide an example data frame of data supposed to match the sampling frame to check if the provided variable names match and whether all strata in the data appear in the sampling frame.

`sampling.frame.file`
: data frame containing the sampling frame. should contain columns "stratum" and "population", otherwise column names must be specified.

Value

returns a new function that takes a data frame as input returns a vector of weights corresponding to each row in the data frame.

Examples

```
# laod data and sampling frames:
mydata<-read.csv("mydata.csv")
mysamplingframe<-read.csv("mysamplingframe.csv")
# create weighting function:
weighting<-weighting_fun_from_samplingframe(sampling.frame = mysamplingframe,
                                  data.stratum.column = "strata_names",
                                  sampling.frame.population.column = "pop",
                                  sampling.frame.stratum.column = "strat_name")
# use weighting function:
mydata$weights<-weighting(mydata)

# this also works on subsets of the data:
mydata_subset<-mydata[1:100,]
subset_weights<- weighting(mydata)
```

Index