Package 'GREGWT'

November 24, 2014

Type Package

Title Implements the GREGWT algorithm in R.

Version 1.4

Depends MASS

Date 2014-11-21

Author M. Esteban Munoz H.

Maintainer M. Esteban Munoz H. <emunozh@gmail.com>

Description Implements the GREGWT (Generalised Regression and Weighting of sample survey results) algorithm in the R.
The algorithm implemented in this package is based on the example presented in:
Rahman, A. (2009). Small Area Estimation Through Spatial Microsimulation Models. In 2nd International Microsimulation Association Conference . Ottawa, Canada.

License GPL-2

R topics documented:

GREGWT *GREGWT algorithm implemented in R*

Description

Implements the GREGWT (Generalised Regression and Weighting of sample survey results) algorithm in the R. The algorithm implemented in this package is based on the example presented in: Rahman, A. (2009). Small Area Estimation Through Spatial Microsimulation Models. In 2nd International Microsimulation Association Conference . Ottawa, Canada.

Details

Package:	GREGWT
Type:	Package
Version:	1.4
Date:	2014-11-24
License:	GPL-2

Implementation of GREGWT in R.

```
Weights.New = GREGWT(X, dx, Tx, group='HHid', bounds=c(0, Inf))
```

Where

1. `X` is the sample, formated either as a `matrix` or as a `data.frame`

2. `dx` are the initial weights formated as a vector

3. `Tx` are the true population totals

4. (Optional) `group` can be set to define one of the columns of `X` to set a grouping parameter (e.g. households id's)

5. (Optional) `bounds` sets the truncation bounds as `c(L, U)`. Default values are: `c(-Inf, Inf)`

6. (Optional) `epsilon` defining the convergence criterion. Default is set to: `epsilon = 0.001`.

7. (Optional) `max.iter` defining the maximum number of iterations. Default is set to: `max.iter = 10`.

Author(s)

M. Esteban Munoz H. Maintainer: emunozh@gmail.com

Examples

```
############################################################
## Simple example
############################################################

# load the R GREGWT library
library(GREGWT)

## Data ######
# Variables:
#   1. age          (1=16-30 years and 0='otherwise')
#   2. sex          (1=female and 0=male)
#   3. employment   (1=unemployed and 0= 'otherwise')
#   4. income       (in real unit values 0, 1, 2, 3, 4 and 5)
#   5. location     (1= rural and 0= urban)
X <- data.frame(
    age =        c(1,1,0,1,0,0,0,1,0,1,0,1,1,0,0,0,1,0,0,1,0,0,1,0,1),
    sex =        c(1,0,0,1,1,0,0,0,1,0,1,1,0,0,1,0,0,1,0,0,0,0,0,1,1,0),
    employment = c(0,1,1,1,0,1,0,1,0,0,1,1,1,1,1,0,1,0,1,1,0,1,0,1,0),
    income =     c(0,3,2,5,0,1,0,4,0,0,1,3,2,5,4,0,3,0,2,4,0,5,0,1,0),
    location =   c(0,1,1,0,1,0,1,0,1,1,0,1,1,1,0,0,1,0,1,0,1,1,0,0,1))

# Initial weights
dx <- c(4,5,6,5,3,4,6,4,5,3,5,4,3,6,4,5,6,3,6,4,5,3,5,4,3)

# True population totals
Tx <- data.frame(age=50,sex=45,employment=70,income=200,location=65)
```

```
# Get new weights with GREGWT
Simulation.Data <- prepareData(X, Tx)
Weights.GREGWT = GREGWT(Simulation.Data$X, dx,
                        Simulation.Data$Tx, bounds=c(0,Inf))
summary(Weights.GREGWT)
print(Weights.GREGWT)

X["Initial.Weights"] <- dx
X["Final.Weights"] <- Weights.GREGWT$Final.Weights

# Create 3 random synthetic populations with new weights
pop <- Synthetize(Weights.GREGWT, 50, random.seed=11111, method=c("random", 3))
View(pop)

# Get the 'best' random synthetic populations out of 10 iterations with new weights
pop <- Synthetize(Weights.GREGWT, 50, random.seed=11111, method=c("best", 10),
                  benchmarks=Tx)
View(pop)
```

GREGWT *Main function to estimate new weights.*

Description

This function estimates new weights given: (a) a matrix containing a survey information from a sample of the population; (b) the initial weights for this sample; and (c) the 'true' population totals to which we aim to re weight the sample to.

Usage

```
GREGWT(X, dx, Tx, ...)
```

Arguments

X	is the sample, formated either as a `matrix` or as a `data.frame`
dx	are the initial weights formated as a `vector`
Tx	are the true population totals
...	
group	can be set to define one of the columns of `X` to set a grouping parameter (e.g. households id's). Default value is: `group = FALSE`
bounds	sets the truncation bounds as `c(L, U)`. Default values are: `c(-Inf, Inf)`
epsilon	defining the convergence criterion. Default is set to: `epsilon = 0.001`
max.iter	defining the maximum number of iterations. Default is set to: `max.iter = 10`
X.input	Defines the original input data. Default is set to: `X.input = FALSE`

Value

Input.Weights	these are the given weights dx
Final.Weights	estimated weights for the given sample

Author(s)

M. Esteban Munoz H. emunozh@gmail.com

`prepareData` *A simple function to prepare the data for simulation.*

Description

This function will: (1) group the data by a specified attribute (needed for an integrated re weighting); (2) remove data columns where all values are either 1 or 0; and (3) check for collinearity in the dataset.

Usage

```
prepareData(X, Tx, ...)
```

Arguments

`X`	is the population sample
`Tx`	are the true population totals
`...`	
`cor.lim`	sets the correlation factor to be used as a limit
`group`	can be set to define one of the columns of `X` to set a grouping parameter (e.g. households id's). Default value is: `group = FALSE`

Value

`X`	formated `X`
`Tx`	formated `Tx`

Author(s)

M. Esteban Munoz H. emunozh@gmail.com

`Synthetize` *Creates a synthetic population with the new weights*

Description

This fuction creates a synthetic population given: (a) the new estimated weights; and (b) the desire population size.

Usage

```
Synthetize(gregwt.object, pop.size)
```

Arguments

`gregwt.object`	resulting object form the use of function: `GREGWT`
`pop.size`	desire population size (int)

Value

`X`	returns a matrix with the original `X` stucture.

Author(s)

M. Esteban Munoz H. emunozh@gmail.com

Index