Package 'anesrake'

October 12, 2022

Version 0.80

Date 2018-04-27

Title ANES Raking Implementation

Author Josh Pasek [aut, cre]

Maintainer Josh Pasek <josh@joshpasek.com>

Depends Hmisc, weights

Description Provides a comprehensive system for selecting variables and weighting data to match the specifications of the American National Election Studies. The package includes methods for identifying discrepant variables, raking data, and assessing the effects of the raking algorithm. It also allows automated re-raking if target variables fall outside identified bounds and allows greater user specification than other available raking algorithms. A variety of simple weighted statistics that were previously in this package (version .55 and earlier) have been moved to the package 'weights.'

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-28 09:23:18 UTC

R topics documented:

anes04 *Demographic Data From 2004 American National Election Studies (ANES)*

Description

A dataset containing demographic data from the 2004 American National Election Studies. The data include 5 variables: "female" (A Logical Variable Indicating Sex), "age" (Numerically Coded, Ranging From 18 to a Topcode of 90), "educats" (5 Education Categories corresponding to 1-Less than A High School Degree, 2-High School Gradutate, 3-Some College, 4-College Graduate, 5-Post College Education), "racecats" (6 Racial Categories), and "married" (A Logical Variable Indicating the Respondent's Marital Status, with one point of missing data). Dataset is designed show how production of survey weights works in practice.

Usage

```
data(anes04)
```

Format

The format is: chr "anes04"

Source

http://www.electionstudies.org

anesrake *Function to perform full ANES variable selection and weighting.*

Description

anesrake takes a list of variables and target values and determines how they should be weighted to match the procedures outlined in DeBell and Krosnick, 2009. It then performs raking to develop weights for the variables selected such that they match the targets provided.

Usage

```
anesrake(inputter, dataframe, caseid, weightvec = NULL,
cap = 5, verbose = FALSE, maxit = 1000, type = "pctlim",
pctlim = 5, nlim = 5, filter = 1, choosemethod = "total",
iterate = TRUE, convcrit = 0.01, force1=TRUE, center.baseweights=TRUE)
```

Arguments

inputter	The inputter object should contain a list of all target values for the raking procedure. Each list element in inputter should be a vector corresponding to the weighting targets for a single variable. Hence, the vector enumerating the weighting targets for a variable with 2 levels should be of length 2, while a vector enumerating the weighting targets for a variable with 5 levels should be of length 5. List elements in inputter should be named according to the variable that they will match in the corresponding dataset. Hence, a list element enumerating the proportion of the sample that should be of each gender should be labeled "female" if the variable in dataframe is also titled "female." inputter elements must be vectors and can be of class numeric, or factor and must match the class of the corresponding variable in dataframe. Logical variables in dataframe can be matched to a numeric vector of length 2 and ordered with the TRUE target as the first element and the FALSE target as the second element. Targets for factors must be labeled to match every level present in the dataframe (e.g. a variable with 2 age groups "under40" and "over40" should have elements named "under40" and "over40" respectively). anesrake attempts to conform any unrecognized types of vectors to class(numeric). Weighting targets can be entered either as an N to be reached or as a percent for any given variable. Targets can be either proportions (ideal) or the number of individuals in the population in each target category (N). Totals of greater than 1.5 for any given list element are treated as Ns, while values of less than 1.5 are treated as percentages.
dataframe	The dataframe command identifies a data.frame object of the data to be weighted. The data.frame must contain all of the variables that will be used in the weighting process and those variables must have the same names as are present in the inputter list element.
caseid	The caseid command identifies a unique case identifier for each individual in the dataset. If filters are to be used, the resulting list of weights will be a different length from the overall dataframe. caseid is included in the output so that weights can be matched to the dataset of relevance. caseid must be of a length matching the number of cases in dataframe.
weightvec	weightvec is an optional input if some kind of base weights, stratification correction, or other sampling probability of note that should be accounted for before weighting is conducted. If defined, weightvec must be of a length equivalent to the number of cases in the dataframe. If undefined, weightvec will be automatically seeded with a vector of 1s.
cap	cap defines the maximum weight to be used. cap can be defined by the user with the command cap=x, where x is any value above 1 at which the algorithm will cap weights. If cap is set below 1, the function will return an error. If cap is set between 1 and 1.5, the function will return a warning that the low cap may substantially increase the amount of time required for weighting. In the absence of a user-defined cap, the algorithm defaults to a starting value of 5 in line with DeBell and Krosnick, 2009. For no cap, cap simply needs to be set to an arbitrarily high number. (Note: Capping using the cap command caps at each iteration.)

verbose — Users interested in seeing the progress of the algorithm can set verbose to equal TRUE. The algorithm will then inform the user of the progress of each raking and capping iteration.

maxit — Users can set a maximum number of iterations for the function should it fail to converge using maxit=X, where X is the maximum number of iterations. The default is set to 1000.

type — type identifies which manner of variable identification should be used to select weighting variables. Five options are available: type=c("nolim", "pctlim", "nlim", "nmin", "nmax"). If type="nolim", all variables specified in inputter will be included in the weighting procedure. If type="pctlim" (DEFAULT), the variable selection algorithm will assess which variables have distributions that deviate from their targets by more than the amount specified by the pctlim command using the method choosemethod. If type="nlim", the variable selection algorithm will use the number of varibles specified by nlim, choosing the most discrepant variables as identified by the choosemethod command. If type="nmin", the variable selection algorithm will use at least nlim variables, but will include more if additional variables are off by more than pctmin (all identified using choosemethod). If type="nmax", the variable selection algorithm will use no more than nlim variables, but will only use that many variables if at least that many are off by more than pctlim (all identified using choosemethod).

pctlim — pctlim is the discrepancy limit for selection. Variable selection will only select variables that are discrepant by more than the amount specified. pctlim can be specified either in percentage points (5 is 5 percent) or as a decimal (.05 is 5 percent). The algorithm assumes that a decimal is being used if pctlim<1. Hence researchers interested in a discrepancy limit of half a percent would need to use pctlim=.005.

nlim — nlim is the number of variables to be chosen via the variable selection method chosen in choosemethod.

filter — filter is a vector of 1 for cases to be included in weighting and 0 for cases that should not be included. The filter vector must have the same number of cases as the dataframe. In the absence of a user-defined filter, the algorithm defaults to a starting value of 1 (inclusion) for all individuals.

choosemethod — choosemethod is the method for choosing most discrepant variables. Six options are available: choosemethod=c("total", "max", "average", "totalsquared", "maxsquared", "averagesquared"). If choosemethod="total", variable choice is determined by the sum of the differences between actual and target values for each prospective weighting variable. If choosemethod="max", variable choice is determined by the largest individual difference between actual and target values for each prospective weighting variable. If choosemethod="average", variable choice is determined by the mean of the differences between actual and target values for each prospective weighting variable. If choosemethod="totalsquared", variable choice is determined by the sum of the squared differences between actual and target values for each prospective weighting variable. If choosemethod="maxsquared", variable choice is determined by the largest squared difference between actual and target values for each prospective weighting variable (note that this is identical to choosemethod="max" if the selection type is nlim). If choosemethod="averagesquared",

variable choice is determined by the mean of the squared differences between actual and target values for each prospective weighting variable.

`iterate` `iterate` is a logical variable for how raking should proceed if `type=c("pctlim", "nmin", "nmax")` conditions. If `iterate=TRUE`, `anesrake` will check whether any variables that were not used in raking deviate from their targets by more than `pctlim` percent. When this is the case, raking will be rerun using the raked weights as seeds (`weightvec`) with additional varibles that meet this qualification after raking included as well. For the `type="nmax"` condition, this will only occur if `nlim` has not been met.

`convcrit` `convcrit` is the criterion for convergence. The raking algorithm is determined to have converged when the most recent iteration represents less than a `convcrit` percentage improvement over the prior iteration.

`force1` `force1` ensures that the categories of each raking variable sum to 1. To do so, the target in inputter for each variable is divided by the sum of the targets for that category.

`center.baseweights` `center.baseweights` forces the initial baseweight to mean to 1 if true (the default setting).

Value

A list object of `anesrake` has the following elements:

`weightvec`	Vector of weights From raking algorithm
`type`	Type of variable selection used (identical to specified `type`)
`caseid`	Case IDs for final weights – helpful for matching `weightvec` to cases if a filter is used
`varsused`	List of variables selected for weighting
`choosemethod`	Method for choosing variables for weighting (identical to specified `choosemethod`)
`converge`	Notes whether full convergence was achieved, algorithm failed to converge because convergence was not possible, or maximum iterations were reached
`nonconvergence`	Measure of remaining discrepancy from benchmarks if convergence was not achieved
`targets`	`inputter` from above, a list of the targets used for weighting
`dataframe`	Copy of the original `dataframe` used for weighting (`filter` variable applied if specified)
`iterations`	Number of iterations required for convergence (or non-convergence) of final model
`iterate`	Copy of `iterate` from above

Author(s)

Josh Pasek, Assistant Professor of Communication Studies at the University of Michigan (www.joshpasek.com).

References

DeBell, M. and J.A. Krosnick. (2009). Computing Weights for American National Election Study
Survey Data, ANES Technical Report Series, No. nes012427. Available from: ftp://ftp.electionstudies.org/ftp/nes/bibliograph

Examples

```
data("anes04")

anes04$caseid <- 1:length(anes04$age)

anes04$agecats <- cut(anes04$age, c(0, 25,35,45,55,65,99))
levels(anes04$agecats) <- c("age1824", "age2534", "age3544",
          "age4554", "age5564", "age6599")

marriedtarget <- c(.4, .6)

agetarg <- c(.10, .15, .17, .23, .22, .13)
names(agetarg) <- c("age1824", "age2534", "age3544",
          "age4554", "age5564", "age6599")

targets <- list(marriedtarget, agetarg)

names(targets) <- c("married", "agecats")

outsave <- anesrake(targets, anes04, caseid=anes04$caseid,
          verbose=TRUE)

caseweights <- data.frame(cases=outsave$caseid, weights=outsave$weightvec)

summary(caseweights)

summary(outsave)
```

---

anesrakefinder *Function to determine what variables should be used for weighting.*

Description

anesrake takes a list of variables and target values and determines which variables should be used for weighting in accordance with DeBell and Krosnick, 2009. Used as part of anesrake.

Usage

```
anesrakefinder(inputter, dataframe, weightvec = NULL,
choosemethod = "total")
```

Arguments

inputter	The inputter object should contain a list of all target values for the raking procedure. Each list element in inputter should be a vector corresponding to the weighting targets for a single variable. Hence, the vector enumerating the weighting targets for a variable with 2 levels should be of length 2, while a vector enumerating the weighting targets for a variable with 5 levels should be of length 5. List elements in inputter should be named according to the variable that they will match in the corresponding dataset. Hence, a list element enumerating the proportion of the sample that should be of each gender should be labeled "female" if the variable in dataframe is also titled "female." inputter elements must be vectors and can be of class numeric, or factor and must match the class of the corresponding variable in dataframe. Logical variables in dataframe can be matched to a numeric vector of length 2 and ordered with the TRUE target as the first element and the FALSE target as the second element. Targets for factors must be labeled to match every level present in the dataframe (e.g. a variable with 2 age groups "under40" and "over40" should have elements named "under40" and "over40" respectively). anesrake attempts to conform any unrecognized types of vectors to class(numeric). Weighting targets can be entered either as an N to be reached or as a percent for any given variable. Targets can be either proportions (ideal) or the number of individuals in the population in each target category (N). Totals of greater than 1.5 for any given list element are treated as Ns, while values of less than 1.5 are treated as percentages.
dataframe	The dataframe command identifies a data.frame object of the data to be weighted. The data.frame must contain all of the variables that will be used in the weighting process and those variables must have the same names as are present in the inputter list element.
weightvec	weightvec is an optional input if some kind of base weights, stratification correction, or other sampling probability of note that should be accounted for before weighting is conducted. If defined, weightvec must be of a length equivalent to the number of cases in the dataframe. If undefined, weightvec will be automatically seeded with a vector of 1s.
choosemethod	choosemethod is the method for choosing most discrepant variables. Six options are available: choosemethod=c("total", "max", "average", "totalsquared", "maxsquared", "averagesquared"). If choosemethod="total", variable choice is determined by the sum of the differences between actual and target values for each prospective weighting variable. If choosemethod="max", variable choice is determined by the largest individual difference between actual and target values for each prospective weighting variable. If choosemethod="average", variable choice is determined by the mean of the differences between actual and target values for each prospective weighting variable. If choosemethod="totalsquared", variable choice is determined by the sum of the squared differences between actual and target values for each prospective weighting variable. If choosemethod="maxsquared", variable choice is determined by the largest squared difference between actual and target values for each prospective weighting variable (note that this is identical to choosemethod="max" if the selection type is nlim). If choosemethod="averagesquared", variable choice is determined by the mean of the squared differences between

actual and target values for each prospective weighting variable.

Value

Returns a vector of variable names and discrepancies via the method chosen in `choosemethod`.

Author(s)

Josh Pasek, Assistant Professor of Communication Studies at the University of Michigan (www.joshpasek.com).

`discrep` *Function to determine the discrepancy for each level of a variable from targets.*

Description

Finds the discrepancy between the proportion of data in each level of a weighted vector and a set of targets for each level of that same vector. Used as part of `anesrake`.

Usage

```
discrep(datavec, targetvec, weightvec)
```

Arguments

`datavec`	Vector of values for a particular variable.
`targetvec`	Vector of targets with a single item per level of that variable.
`weightvec`	Weighting vector to be applied to `datavec`.

Value

Vector of discrepancies at each level.

Author(s)

Josh Pasek, Assistant Professor of Communication Studies at the University of Michigan (www.joshpasek.com).

generaldesigneffect *Calculates a general design effect given weights for a dataset.*

Description

Calculates a general design effect given weights for a dataset.

Usage

```
generaldesigneffect(weightvec)
```

Arguments

weightvec	Vector of weights.

rakelist *Function to perform full ANES weighting on selected variables.*

Description

rakelist takes a list of variables and target values weights a dataset with those variables to match the targets via raking. It is the primary workhorse command of anesrake.

Usage

```
rakelist(inputter, dataframe, caseid, weightvec = NULL, cap = 999999,
 verbose = FALSE, maxit = 1000, convcrit = 0.01)
```

Arguments

inputter	The inputter object should contain a list of all target values for the raking procedure. Each list element in inputter should be a vector corresponding to the weighting targets for a single variable. Hence, the vector enumerating the weighting targets for a variable with 2 levels should be of length 2, while a vector enumerating the weighting targets for a variable with 5 levels should be of length 5. List elements in inputter should be named according to the variable that they will match in the corresponding dataset. Hence, a list element enumerating the proportion of the sample that should be of each gender should be labeled "female" if the variable in dataframe is also titled "female." inputter elements must be vectors and can be of class numeric, or factor and must match the class of the corresponding variable in dataframe. Logical variables in dataframe can be matched to a numeric vector of length 2 and ordered with the TRUE target as the first element and the FALSE target as the second element. Targets for factors must be labeled to match every level present in the dataframe (e.g. a variable with 2 age groups "under40" and "over40" should

have elements named "under40" and "over40" respectively). anesrake attempts to conform any unrecognized types of vectors to class(numeric). Weighting targets can be entered either as an N to be reached or as a percent for any given variable. Targets can be either proportions (ideal) or the number of individuals in the population in each target category (N). Totals of greater than 1.5 for any given list element are treated as Ns, while values of less than 1.5 are treated as percentages.

dataframe — The dataframe command identifies a data.frame object of the data to be weighted. The data.frame must contain all of the variables that will be used in the weighting process and those variables must have the same names as are present in the inputter list element.

caseid — The caseid command identifies a unique case identifier for each individual in the dataset. If filters are to be used, the resulting list of weights will be a different length from the overall dataframe. caseid is included in the output so that weights can be matched to the dataset of relevance. caseid must be of a length matching the number of cases in dataframe.

weightvec — weightvec is an optional input if some kind of base weights, stratification correction, or other sampling probability of note that should be accounted for before weighting is conducted. If defined, weightvec must be of a length equivalent to the number of cases in the dataframe. If undefined, weightvec will be automatically seeded with a vector of 1s.

cap — cap defines the maximum weight to be used. cap can be defined by the user with the command cap=x, where x is any value above 1 at which the algorithm will cap weights. If cap is set below 1, the function will return an error. If cap is set between 1 and 1.5, the function will return a warning that the low cap may substantially increase the amount of time required for weighting. In the absence of a user-defined cap, the algorithm defaults to a starting value of 5 in line with DeBell and Krosnick, 2009. For no cap, cap simply needs to be set to an arbitrarily high number. (Note: Capping using the cap command caps at each iteration.)

verbose — Users interested in seeing the progress of the algorithm can set verbose to equal TRUE. The algorithm will then inform the user of the progress of each raking and capping iteration.

maxit — Users can set a maximum number of iterations for the function should it fail to converge using maxit=X, where X is the maximum number of iterations. The default is set to 1000.

convcrit — convcrit is the criterion for convergence. The raking algorithm is determined to have converged when the most recent iteration represents less than a convcrit percentage improvement over the prior iteration.

Value

A list object of rakelist has the following elements:

weightvec — Vector of weights From raking algorithm

caseid — Case IDs for final weights – helpful for matching weightvec to cases if a filter is used

iterations	Number of iterations required for convergence (or non-convergence) of final model
nonconvergence	Measure of remaining discrepancy from benchmarks if convergence was not achieved
converge	Notes whether full convergence was achieved, algorithm failed to converge because convergence was not possible, or maximum iterations were reached
varsused	List of variables selected for weighting
targets	inputter from above, a list of the targets used for weighting
dataframe	Copy of the original dataframe used for weighting (filter variable applied if specified)

Author(s)

Josh Pasek, Assistant Professor of Communication Studies at the University of Michigan (www.joshpasek.com).

---

weightassess *Assessment of Weighting*

Description

Shows weighted data on specified variables compared to targets and baseweights.

Usage

```
weightassess(inputter, dataframe, weightvec, prevec = NULL)
```

Arguments

inputter	The inputter object should contain a list of all target values for the raking procedure. Each list element in inputter should be a vector corresponding to the weighting targets for a single variable. Hence, the vector enumerating the weighting targets for a variable with 2 levels should be of length 2, while a vector enumerating the weighting targets for a variable with 5 levels should be of length 5. List elements in inputter should be named according to the variable that they will match in the corresponding dataset. Hence, a list element enumerating the proportion of the sample that should be of each gender should be labeled "female" if the variable in dataframe is also titled "female." inputter elements must be vectors and can be of class numeric, or factor and must match the class of the corresponding variable in dataframe. Logical variables in dataframe can be matched to a numeric vector of length 2 and ordered with the TRUE target as the first element and the FALSE target as the second element. Targets for factors must be labeled to match every level present in the dataframe (e.g. a variable with 2 age groups "under40" and "over40" should have elements named "under40" and "over40" respectively). anesrake attempts to conform any unrecognized types of vectors to class(numeric). Weighting targets can be entered either as an N to be reached or as a percent for any given

variable. Targets can be either proportions (ideal) or the number of individuals in the population in each target category (N). Totals of greater than 1.5 for any given list element are treated as Ns, while values of less than 1.5 are treated as percentages.

`dataframe` The `dataframe` command identifies a `data.frame` object of the data to be weighted. The data.frame must contain all of the variables that will be used in the weighting process and those variables must have the same names as are present in the `inputter` list element.

`weightvec` `weightvec` is a vector of final weights that are to be assessed.

`prevec` `prevec` is an optional input if some kind of base weights, stratification correction, or other sampling probability of note that should be accounted for before weighting is conducted. If defined, `prevec` must be of a length equivalent to the number of cases in the `dataframe`. If undefined, `prevec` will be automatically seeded with a vector of 1s.

Value

Prints out a list of all levels of all variables named in inputter. For each variable, shows values weighted with prevec, weightvec, and the targets and assesses discrepancies for each.

Author(s)

Josh Pasek, Assistant Professor of Communication Studies at the University of Michigan (www.joshpasek.com).

Index