Package 'Rweights3'

April 28, 2018

Type Package

Title Rweights3 -- Poststratification Weighting

Version 1.0

Date 2014-04-05

Author Loren Collingwood

Maintainer Who to complain to <loren.collingwood@ucr.edu>

Depends R (>= 3.0), survey

Description More about what it does (maybe more than one line)

License What license is it under?

R topics documented:

`Rweights-package` *Functions for weighting survey data*

Description

User provides target proportions, variables, and data. The result is a survey design object and vector of weights. Can be used for iterative raking.

Details

Package:	Rweights
Type:	Package
Version:	1.0
Date:	2011-10-30
License:	GPL-3
LazyLoad:	yes

Author(s)

Loren Collingwood Maintainer: <loren.collingwood@ucr.edu>

References

Thomas Lumley, "Survey Analysis in R" <http://faculty.washington.edu/tlumley/survey/>

`gen_survey_props` *gen_survey_props*

Description

Generates survey proportions based off of (ideally) an unweighted vector.

Usage

```
gen_survey_props(vector)
```

Arguments

`vector`	Numeric vector

Value

Numeric vector of proportions

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
#NOT RUN: READ IN DATA
data (wapoll)
#NOT RUN -- CREATE PROPORTIONS FROM TARGET DATA
age_current <- gen_survey_props(wapoll$age3) #1=18-40,2=41-65,3=66+; age3
age_current
gender_current <- gen_survey_props(wapoll$gender) #0=Male, 1=Female
gender_current
```

iter_weight *iter_weight*

Description

Upon initial "final weighting" if proportions still do not match with X percentage points of the target weight proportions, use iter_weight to rake the data to obtain desired target weight proportions.

Usage

```
iter_weight(weight_props, weight_tabs, variable, survey_weight_obj, dataframe, prop_diff = 0.01)
```

Arguments

weight_props	Object from weight_to result.
weight_tabs	Numeric vector of proportions probably generated using svymean function, and then put into vector format.
variable	Numeric vector. Usually the unweighted variable from the data.
survey_weight_obj	Survey weights generated from previous round of weighting. Probably from survey_weight function – the second output (survey_weight)
dataframe	Data frame object
prop_diff	Proportion difference analyst is willing to weight to. Default is 0.01, or 1%.

Value

survey_design	Survey design object, based on survey package.
survey_weight	Vector of survey weights for each respondent.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References

Thomas Lumley, "Survey Analysis in R" <http://faculty.washington.edu/tlumley/survey/>

Examples

```
#NOT RUN: READ IN DATA
data (wapoll)

#NOT RUN: CREATE WEIGHT TO PROPORTIONS
weight_props_age <- weight_to(c(26.0,39.3,34.7), c("18-40","41-65","66+"))
weight_props_gender <- weight_to(c(49,51), c("Male","Female"))
weight_props_party <- weight_to(c(33.8,28.1,38.1), c("Dem","Rep","Ind"))
weight_props_region <- weight_to(c(53.5,20.6,25.8), c("Reg1-PS","Reg2-EW","Reg3-O"))

#NOT RUN: CREATE PROPORTIONS FROM DATA
age_current <- gen_survey_props(wapoll$age3) #1=18-40,2=41-65,3=66+; age3
```

```
gender_current <- gen_survey_props(wapoll$gender) #0=Male, 1=Female

#NOT RUN: CREATE WEIGHT TO AMOUNT
age_weight_to <- weight_amount(weight_props_age,age_current)
gender_weight_to <- weight_amount(weight_props_gender,gender_current)

#NOT RUN: CREATE AGE*GENDER WEIGHT
age_weight_2 <- variable_weight(wapoll$age3,age_weight_to)
gender_weight_2 <- variable_weight(wapoll$gender,gender_weight_to)
svyd_age_gender <- survey_weight(cbind(age_weight_2,gender_weight_2),dataframe=wapoll)

#NOT RUN: INCLUDE PARTY IN AGE*GENDER WEIGHT
party_tab2 <- svymean(~interaction(wapoll$pid), design=svyd_age_gender$survey_design, na.rm=TRUE)
weight_to_party <- weight_amount(weight_props_party,c(party_tab2[[1]]*100,party_tab2[[2]]*100,party_tab2[[3
weight_party_2 <- variable_weight(wapoll$pid,weight_to_party)

#NOT RUN: WEIGHT CREATED FOR age*gender*party
svyd_age_gender_party <- survey_weight(cbind(svyd_age_gender$survey_weight,weight_party_2),dataframe=wapoll

#NOT RUN: Region Proportions (weighted by (age*gender)*Party)
region_tab2 <- svymean(~interaction(wapoll$region), design=svyd_age_gender_party$survey_design, na.rm=TRUE)
#NOT RUN: INCLUDE REGION IN PARTY*AGE*GENDER WEIGHT
weight_to_region <- weight_amount(weight_props_region,c(region_tab2[[1]]*100,region_tab2[[2]]*100,region_ta
weight_region_2 <- variable_weight(wapoll$region,weight_to_region)

#NOT RUN: CREATE INITIAL "FINAL" WEIGHT
svyd_age_gender_party_region <- survey_weight(cbind(svyd_age_gender_party$survey_weight,weight_region_2),da

#NOT RUN: EXAMINE WEIGHTED TABS
age_tabs <- svymean(~interaction(wapoll$age3), design=svyd_age_gender_party_region$survey_design, na.rm=TRUE
gender_tabs <- svymean(~interaction(wapoll$gender), design=svyd_age_gender_party_region$survey_design, na.rm
party_tabs <- svymean(~interaction(wapoll$pid), design=svyd_age_gender_party_region$survey_design, na.rm=TRU
region_tabs <- svymean(~interaction(wapoll$region), design=svyd_age_gender_party_region$survey_design, na.rm

#NOT RUN: RUN iter_weight FUNCTION FOR AGE
iter <- iter_weight(weight_props = weight_props_age,
weight_tabs = matrix(age_tabs)[,1]*100,
variable = wapoll$age3,
survey_weight_obj = svyd_age_gender_party_region$survey_weight,
dataframe=wapoll
)
age_tabs <- svymean(~interaction(wapoll$age3), design=iter$survey_design, na.rm=TRUE)
tab_look(weight_props_age, matrix(age_tabs)[,1]*100)

#NOT RUN: RUN iter_weight FUNCTION FOR GENDER
iter2 <- iter_weight(weight_props = weight_props_gender,
weight_tabs = matrix(gender_tabs)[,1]*100,
variable = wapoll$gender,
survey_weight_obj = iter$survey_weight,
dataframe=wapoll
)
gender_tabs2 <- svymean(~interaction(wapoll$gender), design=iter2$survey_design, na.rm=TRUE)
tab_look(weight_props_gender, matrix(gender_tabs2)[,1]*100)

#NOT RUN: LOOK AT THE OTHER PROPORTIONS TO MAKE SURE THEY ARE GOOD
age_tabs2 <- svymean(~interaction(wapoll$age3), design=iter2$survey_design, na.rm=TRUE)
tab_look(weight_props_age, matrix(age_tabs2)[,1]*100)
```

```
party_tabs2 <- svymean(~interaction(wapoll$pid), design=iter2$survey_design, na.rm=TRUE)
tab_look(weight_props_party, matrix(party_tabs2)[,1]*100)

region_tabs2 <- svymean(~interaction(wapoll$region), design=iter2$survey_design, na.rm=TRUE)
tab_look(weight_props_region, matrix(region_tabs2)[,1]*100)
```

`survey_weight` *survey_weight*

Description

Creates a survey design object of class list, with survey design and vector of weights as two outputs.

Usage

```
survey_weight(matrix, dataframe, ...)
```

Arguments

`matrix`	`matrix` object, either as a matrix or as a `cbind` object
`dataframe`	`dataframe` object. Probably the data frame with which you are working.
`...`	Arguments passed on from other functions.

Value

`survey_design`	Survey design object, from package `survey`

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References

Thomas Lumley, "Survey Analysis in R" <http://faculty.washington.edu/tlumley/survey/>

Examples

```
#NOT RUN: READ IN DATA
data (wapoll)

#NOT RUN: CREATE WEIGHT TO PROPORTIONS
weight_props_age <- weight_to(c(26.0,39.3,34.7), c("18-40","41-65","66+"))
weight_props_gender <- weight_to(c(49,51), c("Male","Female"))

#NOT RUN: CREATE PROPORTIONS FROM DATA
age_current <- gen_survey_props(wapoll$age3) #1=18-40,2=41-65,3=66+; age3
gender_current <- gen_survey_props(wapoll$gender) #0=Male, 1=Female

#NOT RUN: CREATE WEIGHT TO AMOUNT
age_weight_to <- weight_amount(weight_props_age,age_current)
gender_weight_to <- weight_amount(weight_props_gender,gender_current)
```

```
#NOT RUN: CREATE AGE*GENDER WEIGHT
age_weight_2 <- variable_weight(wapoll$age3,age_weight_to)
gender_weight_2 <- variable_weight(wapoll$gender,gender_weight_to)
svyd_age_gender <- survey_weight(cbind(age_weight_2,gender_weight_2),dataframe=wapoll)
names(svyd_age_gender)
svyd_age_gender$survey_design
summary(svyd_age_gender$survey_weight)
```

---

tab_look *tab_look*

Description

Prints table of "weight to" proportions, actual proportions (from unweighted or weighted data), absolute difference, and percent difference.

Usage

```
tab_look(vector1, vector2)
```

Arguments

vector1	Numeric vector of proportions
vector2	Numeric vector of, ideally, weighted proportions

Value

Matrix object of values, including columns of initial vector of "weight to" proportions, weighted proportions, absolute difference, and percent difference between vector1 and vector2.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
#NOT RUN: READ IN DATA
data (wapoll)

#NOT RUN: CREATE WEIGHT TO PROPORTIONS
weight_props_age <- weight_to(c(26.0,39.3,34.7), c("18-40","41-65","66+"))
weight_props_gender <- weight_to(c(49,51), c("Male","Female"))
weight_props_party <- weight_to(c(33.8,28.1,38.1), c("Dem","Rep","Ind"))
weight_props_region <- weight_to(c(53.5,20.6,25.8), c("Reg1-PS","Reg2-EW","Reg3-O"))

#NOT RUN: CREATE PROPORTIONS FROM DATA
age_current <- gen_survey_props(wapoll$age3) #1=18-40,2=41-65,3=66+; age3
gender_current <- gen_survey_props(wapoll$gender) #0=Male, 1=Female

#NOT RUN: CREATE WEIGHT TO AMOUNT
age_weight_to <- weight_amount(weight_props_age,age_current)
gender_weight_to <- weight_amount(weight_props_gender,gender_current)
```

```
#NOT RUN: CREATE AGE*GENDER WEIGHT
age_weight_2 <- variable_weight(wapoll$age3,age_weight_to)
gender_weight_2 <- variable_weight(wapoll$gender,gender_weight_to)
svyd_age_gender <- survey_weight(cbind(age_weight_2,gender_weight_2),dataframe=wapoll)

#NOT RUN: INCLUDE PARTY IN AGE*GENDER WEIGHT
party_tab2 <- svymean(~interaction(wapoll$pid), design=svyd_age_gender$survey_design, na.rm=TRUE)
weight_to_party <- weight_amount(weight_props_party,c(party_tab2[[1]]*100,party_tab2[[2]]*100,party_tab2[[3
weight_party_2 <- variable_weight(wapoll$pid,weight_to_party)

#NOT RUN: WEIGHT CREATED FOR age*gender*party
svyd_age_gender_party <- survey_weight(cbind(svyd_age_gender$survey_weight,weight_party_2),dataframe=wapoll

#NOT RUN: Region Proportions (weighted by (age*gender)*Party)
region_tab2 <- svymean(~interaction(wapoll$region), design=svyd_age_gender_party$survey_design, na.rm=TRUE)
#NOT RUN: INCLUDE REGION IN PARTY*AGE*GENDER WEIGHT
weight_to_region <- weight_amount(weight_props_region,c(region_tab2[[1]]*100,region_tab2[[2]]*100,region_ta
weight_region_2 <- variable_weight(wapoll$region,weight_to_region)

#NOT RUN: CREATE INITIAL "FINAL" WEIGHT
svyd_age_gender_party_region <- survey_weight(cbind(svyd_age_gender_party$survey_weight,weight_region_2),da

#NOT RUN: EXAMINE WEIGHTED TABS
age_tabs <- svymean(~interaction(wapoll$age3), design=svyd_age_gender_party_region$survey_design, na.rm=TRUE
gender_tabs <- svymean(~interaction(wapoll$gender), design=svyd_age_gender_party_region$survey_design, na.rm
party_tabs <- svymean(~interaction(wapoll$pid), design=svyd_age_gender_party_region$survey_design, na.rm=TRU
region_tabs <- svymean(~interaction(wapoll$region), design=svyd_age_gender_party_region$survey_design, na.rm

#NOT RUN: PUT WEIGHT PROPORTIONS AND NEWLY CALCULATED WEIGHTED PROPORTIONS IN VECTORS AND SEND TO tab_look
weight_props <- c( weight_props_age, weight_props_gender, weight_props_party, weight_props_region)
weighted_data <- c( matrix(age_tabs)[,1]*100, matrix(gender_tabs)[,1]*100, matrix(party_tabs)[,1]*100, matrix
tab_weights <- tab_look(weight_props, weighted_data)
tab_weights
```

`variable_weight` *variable_weight*

Description

Recode an existing variable to its "weight to" proportions

Usage

```
variable_weight(vector, var_weight_to)
```

Arguments

`vector`	Numeric vector from unweighted data.
`var_weight_to`	Numeric vector. Probably object from "weight amount" function.

Value

Numeric vector of weights applied to each respondent in the dataset.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
#NOT RUN: READ IN DATA
data (wapoll)

#NOT RUN: CREATE WEIGHT TO PROPORTIONS
weight_props_age <- weight_to(c(26.0,39.3,34.7), c("18-40","41-65","66+"))
weight_props_gender <- weight_to(c(49,51), c("Male","Female"))

#NOT RUN: CREATE PROPORTIONS FROM DATA
age_current <- gen_survey_props(wapoll$age3) #1=18-40,2=41-65,3=66+; age3
gender_current <- gen_survey_props(wapoll$gender) #0=Male, 1=Female

#NOT RUN: CREATE WEIGHT TO AMOUNT
age_weight_to <- weight_amount(weight_props_age,age_current)
gender_weight_to <- weight_amount(weight_props_gender,gender_current)

#NOT RUN: CREATE AGE*GENDER WEIGHT
age_weight_2 <- variable_weight(wapoll$age3,age_weight_to)
summary(age_weight_2)
gender_weight_2 <- variable_weight(wapoll$gender,gender_weight_to)
summary(gender_weight_2)
```

`wapoll` *wapoll – Washington Poll*

Description

Partial data from 2011 Washington Poll. Only includes variables respnum, age3, gender, pid, region, for weighting.

Usage

```
data(wapoll)
```

Format

A data frame with 825 observations on the following 5 variables.

`respnum` a numeric vector of unique respondent ids.

`age3` a numeric vector, with categories 18-40,41-65,66+.

`gender` a numeric vector, with categories 1=Male, 2=Female.

`pid` a numeric vector, with categories 1=Democrat, 2=Republican, 3=Independent/Other.

`region` a numeric vector, with categories 1=Puget Sound, 2=Eastern Washington, 3=Other.

Source

Washington Poll 2011. Matt Barreto, Director and Loren Collingwood, Senior Researcher.

References

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
data(wapoll)
str(wapoll)
head(wapoll)
```

`weight_amount` *weight_amount*

Description

Creates a weight amount ratio that will be applied to each respondent.

Usage

```
weight_amount(weight_props, survey_props)
```

Arguments

`weight_props`	Numeric vector. Probably object created from weight_to function.
`survey_props`	Numeric vector. Probably object created from gen_survey_props function.

Value

Numeric vector of "weight to" proportions.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
#NOT RUN: READ IN DATA
data (wapoll)

#NOT RUN: CREATE WEIGHT TO PROPORTIONS
weight_props_age <- weight_to(c(26.0,39.3,34.7), c("18-40","41-65","66+"))
weight_props_gender <- weight_to(c(49,51), c("Male","Female"))

#NOT RUN: CREATE PROPORTIONS FROM DATA
age_current <- gen_survey_props(wapoll$age3) #1=18-40,2=41-65,3=66+; age3
gender_current <- gen_survey_props(wapoll$gender) #0=Male, 1=Female

#NOT RUN: CREATE WEIGHT TO AMOUNT
age_weight_to <- weight_amount(weight_props_age,age_current)
age_weight_to
gender_weight_to <- weight_amount(weight_props_gender,gender_current)
gender_weight_to
```

weight_to *weight_to*

Description

Creates proportions (with names) of the variables you want to weight to.

Usage

```
weight_to(vector, names)
```

Arguments

vector	Numeric vector of the proportions. Probably something like c(33,33,34).
names	Character vector of names. Probably something like c("Dem","Rep","Ind")

Value

Numeric vector of proportions with names associated with it.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
#NOT RUN: CREATE WEIGHT TO PROPORTIONS
weight_props_age <- weight_to(c(26.0,39.3,34.7), c("18-40","41-65","66+"))
weight_props_age
weight_props_gender <- weight_to(c(49,51), c("Male","Female"))
weight_props_gender
weight_props_party <- weight_to(c(33.8,28.1,38.1), c("Dem","Rep","Ind"))
weight_props_party
weight_props_region <- weight_to(c(53.5,20.6,25.8), c("Reg1-PS","Reg2-EW","Reg3-O"))
weight_props_region
```

Index