Package ‘calidad’

September 21, 2022

Type Package

Title Assesses the Quality of Estimates Made by Complex Sample Designs

Version 0.2.0


License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Depends R (>= 3.5.0)

Imports rlang, dplyr, purrr, survey, kableExtra, stringr, tidyr, haven

NeedsCompilation no

Author Klaus Lehmann [aut, cre], Ricardo Pizarro [aut], Ignacio Agloni [ctb]

Maintainer Klaus Lehmann <klehmann@fen.uchile.cl>

Repository CRAN

Date/Publication 2022-09-21 20:00:02 UTC

R topics documented:

casen ................................................................. 2
Description

CASEN data for the year 2020. Contains only a few variables.

Usage

casen

Format

dataframe with 185,437 rows y 6 columns

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folio</td>
<td>household id</td>
</tr>
<tr>
<td>sexo</td>
<td>1 = man; 2 = woman</td>
</tr>
<tr>
<td>edad</td>
<td>age</td>
</tr>
<tr>
<td>activ</td>
<td>Economic activity status</td>
</tr>
<tr>
<td>ing_aut_hog</td>
<td>Household Income</td>
</tr>
<tr>
<td>pobreza</td>
<td>poverty status: 1 = extreme poverty, 2 = non-extreme poverty, 3 = non-poverty</td>
</tr>
<tr>
<td>expr</td>
<td>regional sample weights</td>
</tr>
<tr>
<td>estrato</td>
<td>strata</td>
</tr>
<tr>
<td>cod_upm</td>
<td>PSU</td>
</tr>
</tbody>
</table>
create_html

Source


Examples

data(casen)

create_html  Create html table with the results of the evaluation

Description

Create html table with the results of the evaluation

Usage

create_html(table)

Arguments

table  dataframe generated by evaluate function

Value

html table

Examples

library(survey)
library(dplyr)

hogar <- epf_personas %>%
  group_by(folio) %>%
slice(1)
dc <- survey::svydesign(ids = ~varunit, strata = ~varstrat, data = hogar, weights = ~fe)
table <- evaluate(create_prop("ocupado", domains = "zona+sexo", design = dc))
create_mean

Create the inputs to evaluate the quality of mean estimations

Description

create_mean generates a dataframe with the following elements: mean, degrees of freedom, sample size and coefficient of variation. The function allows grouping in several domains.

Usage

create_mean(
  var,
  domains = NULL,
  subpop = NULL,
  design,
  ci = FALSE,
  ess = FALSE,
  ajuste_ene = FALSE,
  standard_eval = FALSE,
  rm.na = FALSE,
  deff = FALSE,
  rel_error = FALSE,
  unweighted = FALSE,
  eclac_input = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>numeric variable within the dataframe.</td>
</tr>
<tr>
<td>domains</td>
<td>domains to be estimated separated by the + character.</td>
</tr>
<tr>
<td>subpop</td>
<td>integer dummy variable to filter the dataframe</td>
</tr>
<tr>
<td>design</td>
<td>complex design created by survey package</td>
</tr>
<tr>
<td>ci</td>
<td>boolean indicating if the confidence intervals must be calculated</td>
</tr>
<tr>
<td>ess</td>
<td>boolean Effective sample size</td>
</tr>
<tr>
<td>ajuste_ene</td>
<td>boolean indicating if an adjustment for the sampling-frame transition period must be used</td>
</tr>
<tr>
<td>standard_eval</td>
<td>boolean Indicating if the function is wrapped inside a function, if TRUE avoid lazy eval errors</td>
</tr>
<tr>
<td>rm.na</td>
<td>boolean Remove NA if it is required</td>
</tr>
<tr>
<td>deff</td>
<td>boolean Design effect</td>
</tr>
<tr>
<td>rel_error</td>
<td>boolean Relative error</td>
</tr>
<tr>
<td>unweighted</td>
<td>boolean Add non weighted count if it is required</td>
</tr>
<tr>
<td>eclac_input</td>
<td>boolean return eclac inputs</td>
</tr>
</tbody>
</table>
**create_prop**

**Value**

dataframe that contains the inputs and all domains to be evaluated

**Examples**

dc <- survey::svydesign(ids = ~varunit, strata = ~varstrat, data = epf_personas, weights = ~fe)
create_mean("gastot_hd", "zona+sexo", design = dc)

create_prop

**Create the inputs to evaluate the quality of proportion estimations**

**Description**

create_prop generates a dataframe with the following elements: sum, degrees of freedom, sample size, standard error and coefficient of variation. The function allows grouping in several domains.

**Usage**

create_prop(
  var,
  denominador = NULL,
  domains = NULL,
  subpop = NULL,
  design,
  ci = FALSE,
  deff = FALSE,
  ess = FALSE,
  ajuste_ene = FALSE,
  rel_error = FALSE,
  log_cv = FALSE,
  unweighted = FALSE,
  standard_eval = FALSE,
  eclac_input = FALSE
)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>numeric variable within the dataframe, is the numerator of the ratio to be calculated.</td>
</tr>
<tr>
<td>denominador</td>
<td>numeric variable within the dataframe, is the denominator of the ratio to be calculated. If the var parameter is dummy, it can be NULL</td>
</tr>
<tr>
<td>domains</td>
<td>domains to be estimated separated by the + character.</td>
</tr>
<tr>
<td>subpop</td>
<td>integer dummy variable to filter the dataframe</td>
</tr>
<tr>
<td>design</td>
<td>complex design created by survey package</td>
</tr>
<tr>
<td>ci</td>
<td>boolean indicating if the confidence intervals must be calculated</td>
</tr>
</tbody>
</table>
create_prop_internal

- **deff**: boolean, Design effect
- **ess**: boolean, Effective sample size
- **ajuste_ene**: boolean, indicating if an adjustment for the sampling-frame transition period must be used
- **rel_error**: boolean, Relative error
- **log_cv**: boolean, logarithmic coefficient of variation
- **unweighted**: boolean, Add non weighted count if it is required
- **standard_eval**: boolean, Indicating if the function is wrapped inside a function, if TRUE avoid lazy eval errors
- **eclac_input**: boolean, return eclac inputs

**Value**

dataframe that contains the inputs and all domains to be evaluated

**Examples**

```r
library(survey)
library(dplyr)

epf <- mutate(epf_personas, gasto_zona1 = if_else(zona == 1, gastot Hd, 0))
dc <- svydesign(ids = ~varunit, strata = ~varstrat, data = epf, weights = ~fe)
old_options <- options()
options(survey.lonely.psu = "certainty")

create_prop(var = "gasto_zona1", denominador = "gastot Hd", design = dc)

enusc <- filter(enusc, Kish == 1)
dc <- svydesign(ids = ~Conglomerado, strata = ~VarStrat, data = enusc, weights = ~Fact_Pers)
options(survey.lonely.psu = "certainty")
create_prop(var = "VP_DC", denominador = "hom_insg_taxi", design = dc)
options(old_options)
```

---

**create_prop_internal**  
*internal function to calculate proportion estimations*

**Description**

internal function to calculate proportion estimations
create_prop_internal

Usage

create_prop_internal(
  var,
  domains = NULL,
  subpop = NULL,
  disenio,
  ci = FALSE,
  deff = FALSE,
  ess = FALSE,
  ajuste_ene = FALSE,
  rel_error = FALSE,
  log_cv = FALSE,
  unweighted = FALSE,
  standard_eval = TRUE,
  rm.na = FALSE,
  env = parent.frame()
)

Arguments

var integer dummy variable within the dataframe
domains domains to be estimated separated by the + character.
subpop integer dummy variable to filter the dataframe
disenio complex design created by survey package
ci boolean indicating if the confidence intervals must be calculated
deff boolean Design effect
ess boolean Effective sample size
ajuste_ene boolean indicating if an adjustment for the sampling-frame transition period must be used
rel_error boolean Relative error
log_cv boolean indicating if the log cv must be returned
unweighted boolean Add non weighted count if it is required
standard_eval boolean indicating if the function is inside another function, by default it is TRUE, avoid problems with lazy eval.
rm.na boolean indicating if NA values must be removed
env parent environment to get some variables

Value
dataframe that contains the inputs and all domains to be evaluated
create_ratio_internal  *internal function to calculate ratios estimations*

**Description**

internal function to calculate ratios estimations

**Usage**

```r
create_ratio_internal(
  var,
  denominador,
  domains = NULL,
  subpop = NULL,
  disenio,
  ci = FALSE,
  deff = FALSE,
  ess = FALSE,
  ajuste_ene = FALSE,
  unweighted = FALSE,
  rel_error = FALSE,
  rm.na = FALSE
)
```

**Arguments**

- `var` numeric variable within the dataframe, is the numerator of the ratio to be calculated.
- `denominador` numeric variable within the dataframe, is the denominator of the ratio to be calculated.
- `domains` domains to be estimated separated by the + character.
- `subpop` integer dummy variable to filter the dataframe
- `disenio` complex design created by survey package
- `ci` boolean indicating if the confidence intervals must be calculated
- `deff` boolean Design effect
- `ess` boolean Effective sample size
- `ajuste_ene` boolean indicating if an adjustment for the sampling-frame transition period must be used
- `unweighted` boolean Add non weighted count if it is required
- `rel_error` boolean Relative error
- `rm.na` boolean indicating if NA values must be removed

**Value**

dataframe that contains the inputs and all domains to be evaluated
create_size  Create the inputs to evaluate the quality of total estimations

Description

create_size generates a dataframe with the following elements: sum, degrees of freedom, sample size and coefficient of variation. The function allows grouping in several domains.

Usage

create_size(
  var,
  domains = NULL,
  subpop = NULL,
  design,
  ci = FALSE,
  ess = FALSE,
  ajuste_ene = FALSE,
  standard_eval = FALSE,
  rm_na = FALSE,
  deff = FALSE,
  rel_error = FALSE,
  unweighted = FALSE,
  df_type = "ine",
  eclac_input = FALSE
)

Arguments

var numeric variable within the dataframe. When the domain parameter is not used, it is possible to include more than one variable using the + separator. When a value is introduced in the domain parameter, the estimation variable must be a dummy variable.

domains domains to be estimated separated by the + character.

subpop integer dummy variable to filter the dataframe

design complex design created by survey package

ci boolean indicating if the confidence intervals must be calculated

ess boolean Effective sample size

ajuste_ene boolean indicating if an adjustment for the sampling-frame transition period must be used

standard_eval boolean Indicating if the function is wrapped inside a function, if TRUE avoid lazy eval errors

rm_na boolean Remove NA if it is required

deff boolean Design effect
rel_error boolean Relative error
unweighted boolean Add non weighted count if it is required
df_type string Use degrees of freedom calculation approach from INE Chile or CEPAL, by default "ine".
eclac_input boolean return eclac inputs

Value
dataframe that contains the inputs and all domains to be evaluated

Examples
dc <- survey::svydesign(ids = ~varunit, strata = ~varstrat, data = epf_personas, weights = ~fe)
create_size("ocupado", "zona+sexo", design = dc)

dc <- survey::svydesign(ids = ~varunit, strata = ~varstrat, data = epf_personas, weights = ~fe)
create_size("ocupado", "zona+sexo", design = dc)

create_total Create the inputs to evaluate the quality of the sum of continuous variables

Description
create_total generates a dataframe with the following elements: sum, degrees of freedom, sample size and coefficient of variation. The function allows grouping in several domains.

Usage
create_total(
  var,
  domains = NULL,
  subpop = NULL,
  design,
  ci = FALSE,
  ess = FALSE,
  ajuste_ene = FALSE,
  standard_eval = FALSE,
  rm.na = FALSE,
  deff = FALSE,
  rel_error = FALSE,
  unweighted = FALSE,
  eclac_input = FALSE
)
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>numeric variable within the dataframe.</td>
</tr>
<tr>
<td>domains</td>
<td>domains to be estimated separated by the + character.</td>
</tr>
<tr>
<td>subpop</td>
<td>integer dummy variable to filter the dataframe</td>
</tr>
<tr>
<td>design</td>
<td>complex design created by survey package</td>
</tr>
<tr>
<td>ci</td>
<td>boolean indicating if the confidence intervals must be calculated</td>
</tr>
<tr>
<td>ess</td>
<td>boolean Effective sample size</td>
</tr>
<tr>
<td>ajuste_ene</td>
<td>boolean indicating if an adjustment for the sampling-frame transition period must be used</td>
</tr>
<tr>
<td>standard_eval</td>
<td>boolean Indicating if the function is wrapped inside a function, if TRUE avoid lazy eval errors</td>
</tr>
<tr>
<td>rm.na</td>
<td>boolean Remove NA if it is required</td>
</tr>
<tr>
<td>deff</td>
<td>boolean Design effect</td>
</tr>
<tr>
<td>rel_error</td>
<td>boolean Relative error</td>
</tr>
<tr>
<td>unweighted</td>
<td>boolean Add non weighted count if it is required</td>
</tr>
<tr>
<td>eclac_input</td>
<td>boolean return eclac inputs</td>
</tr>
</tbody>
</table>

Value
dataframe that contains the inputs and all domains to be evaluated

Examples
dc <- survey::svydesign(ids = ~varunit, strata = ~varstrat, data = epf_personas, weights = ~fe)
create_total("gastot_hd", "zona+sexo", subpop = "ocupado", design = dc)
Format

dataframe with 87.842 rows and 7 columns

- **sexo** 1 = man; 2 = woman
- **region** region
- **cae_especifico** Economic activity status
- **fe** sample weights
- **varunit** PSU
- **varstrat** strata
- **fdt** It shows if the person belongs to labour force: 1 = yes; 0 = no
- **ocupado** 1 = employed; 0 = non-employed
- **desocupado** 1 = non-employed; 0 = employed

Source

https://www.ine.cl/estadisticas/sociales/mercado-laboral/ocupacion-y-desocupacion

Examples

```
data(ene)
```

| enusc | Encuesta Nacional Urbana de Seguridad ciudadana 2019 - ENUSC 2019 |

Description

ENUSC data for the year 2019. Contains only a few variables.

Usage

```r
enusc
```

Format

dataframe with 24.465 rows and 22 columns

- **rph_sexo** 1 = man; 2 = woman
- **region** 16 regions
- **Fact_Pers** person sample weights
- **Fact_Hog** household sample weights
- **Conglomerado** PSU
- **VarStrat** strata
**VP_DC**  Individual victimization. It works combined with Fact_Pers

**VA_DC**  Household victimization. It works combined with Fact_Hog

**rph_edad**  age

**P3_1_1**  Perception of increased crime in the country. It works combined with Fact_Pers

**P8_1_1**  Cause of increased crime in the neighborhood. It works combined with Fact_Pers

**muj_insg_taxi**  Female perception of insecurity inside taxis. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**hom_insg_taxi**  Male perception of insecurity inside taxis. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**muj_insg_micro**  Female perception of insecurity inside buses. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**hom_insg_micro**  Male perception of insecurity inside buses. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**muj_insg_centr.com**  Female perception of insecurity inside malls. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**hom_insg_centr.com**  Male perception of insecurity inside malls. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**muj_insg_loc.col**  Female perception of insecurity public transport. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**hom_insg_loc.col**  Male perception of insecurity public transport. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**muj_insg_barrio**  Female perception of insecurity neighborhood. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**hom_insg_barrio**  Male perception of insecurity neighborhood. Variable elaborated with variables P9 and rph_sexo. It works combined with Fact_Pers

**Source**


**Examples**

data(enusc)
Description

Reduced version of the VIII EPF database. Contains some sociodemographic variables and the necessary information to work with complex design.

Usage

epf_personas

Format

dataframe compuesto por 48.308 observaciones y 8 variables

- **sexo**: 1 = male; 2 = female
- **zona**: 1 = metropolitan area; 2 = rest of the regional capitals
- **ecivil**: marital status
- **fe**: sample weights
- **varunit**: PSU
- **varstrat**: strata
- **gastot_hd**: household expenditure
- **ocupado**: 1 = employed; 0 = non-employed

Source

https://www.ine.cl/estadisticas/sociales/ingresos-y-gastos/encuesta-de-presupuestos-familiares

Examples

```r
data(epf_personas)
```
evaluate  

**Evaluate the quality of mean estimations**

**Description**

*evaluate* evaluates the quality of mean estimation using the methodology created by INE Chile, which considers sample size, degrees of freedom and coefficient of variation.

**Usage**

```r
evaluate(table, publish = FALSE, scheme = c("chile", "cepal"), ...)
```

**Arguments**

- `table`: dataframe created by `crear_insumos_media`
- `publish`: boolean indicating if the evaluation of the complete table must be added. If it is `TRUE`, the function adds a new column to the dataframe.
- `scheme`: string variable, default scheme is "chile" which refers to the evaluation protocol proposed by INE Chile. The alternative is "cepal" to use the CEPAL protocol.
- `...`: the list of cepal parameters. The complete list of parameters is
  1. General Parameters
     - `df`: degrees of freedom. default: 9
     - `n`: sample size. default ine scheme is 60. default cepal scheme: 100
  2. INE parameters
     - `cv_lower_ine`: lower limit for cv. default: 0.15
     - `cv_upper_ine`: upper limit for cv. default: 0.3
  3. CEPAL parameters
     - `cv_cepal`: limit for cv. default: 0.2
     - `ess`: effective sample size. default: 140
     - `unweighted`: unweighted count. default: 50

**Value**

dataframe with all the columns included in the input table, plus a new column containing a label indicating the evaluation of each estimation: reliable, bit reliable or unreliable.

**Examples**

```r
dc <- survey::svydesign(ids = ~varunit, strata = ~varstrat, data = epf_personas, weights = ~fe)
evaluate(create_mean("gastot_hd", domains = "zona+sexo", design = dc))
```
**get_cv**  
*Get the coefficient of variation*

**Description**

Receive a table created with survey and return the coefficient of variation for each cell

**Usage**

```
get_cv(table, design, domains)
```

**Arguments**

- **table**: dataframe with results
- **design**: design
- **domains**: vector with domains

**Value**

dataframe with results including including CV

---

**get_df**  
*Get degrees of freedom*

**Description**

Receive data and domains. Returns a data frame with the psu, strata and df for each cell

**Usage**

```
get_df(data, domains, df_type = "cepal")
```

**Arguments**

- **data**: dataframe
- **domains**: string with domains
- **df_type**: string Use degrees of freedom calculation approach from INE Chile or CEPAL, by default "ine".

**Value**

dataframe with results including degrees of freedom
get_survey_table \hspace{1cm} Calculates multiple estimations. Internal wrapper for survey package

Description
Generates a table with estimates for a given aggregation

Usage
get_survey_table(
  var,
  domains,
  complex_design,
  estimation = "mean",
  env = parent.frame(),
  fun,
  denom = NULL
)

Arguments

var \hspace{1cm} string objective variable
domains \hspace{1cm} domains
complex_design \hspace{1cm} design from survey
estimation \hspace{1cm} string indicating if the mean must be calculated
env \hspace{1cm} environment parent frame
fun \hspace{1cm} function required regarding the estimation
denom \hspace{1cm} denominator. This parameter works for the ratio estimation

Value
dataframe containing main results from survey

quadratic \hspace{1cm} Calcula el valor de una función cuadrática

Description
quadratic returns the output of a particular function created by INE Chile, which is evaluated at the value of the estimated proportion from a sample. If the output of the function is higher than the standard error, it is interpreted as a signal that the estimation is not reliable.
standardize_design_variables

Usage
quadratic(p)

Arguments
p numeric vector with the values of the estimations for proportions

Value
numeric vector

standardize_columns standardize and sort column names

Description
Receive the survey table in raw state and sort it

Usage
standardize_columns(data, var, denom)

Arguments
data dataframe with results
var string with the objective variable
denom denominator

Value
dataframe with standardized data

standardize_design_variables

Description
Rename design variables, so we can use the later

Usage
standardize_design_variables(design)
standardize_design_variables

Arguments

  design  dataframe

Value

  design survey
Index

* datasets
  
  casen, 2
  ene, 11
  enusc, 12
  epf_personas, 14

  casen, 2
  create_html, 3
  create_mean, 4
  create_prop, 5
  create_prop_internal, 6
  create_ratio_internal, 8
  create_size, 9
  create_total, 10

  ene, 11
  enusc, 12
  epf_personas, 14
  evaluate, 15

  get_cv, 16
  get_df, 16
  get_survey_table, 17

  quadratic, 17

  standardize_columns, 18
  standardize_design_variables, 18