Package 'surveytable'

January 18, 2024

Title Formatted Survey Estimates

Version 0.9.2

Description Short and understandable commands that generate tabulated, formatted, and rounded survey estimates. Mostly a wrapper for the 'survey' package (Lumley (2004) <doi:10.18637/jss.v009.i08> <https://CRAN.R-project.org/package=survey>) that implements the National Center for Health Statistics (NCHS) presentation standards (Parker et al. (2017) <https://www.cdc.gov/nchs/data/series/sr_02/sr02_175.pdf>, Parker et al. (2023) <doi:10.15620/cdc:124368>).

Date/Publication 2024-01-18 20:50:02 UTC

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 2.10)

LazyData true

LazyDataCompression bzip2

Imports assertthat, forcats, huxtable, magrittr, survey

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://cdcgov.github.io/surveytable/,
https://github.com/CDCgov/surveytable

Language en-US

NeedsCompilation no

Author Alex Strashny [aut, cre] (<https://orcid.org/0000-0002-6408-7745>)

Maintainer Alex Strashny <AStrashny@cdc.gov>

Repository CRAN

R topics documented:

`namcs2019sv` *Selected variables from the National Ambulatory Medical Care Survey (NAMCS) 2019 Public Use File (PUF)*

Description

Selected variables from a data system of visits to office-based physicians. Note that the unit of observation is visits, not patients - this distinction is important since a single patient can make multiple visits.

Usage

```
namcs2019sv

namcs2019sv_df
```

Format

An object of class `survey.design2` (inherits from `survey.design`) with 8250 rows and 33 columns.

An object of class `data.frame` with 8250 rows and 33 columns.

Details

`namcs2019sv_df` is a data frame.

`namcs2019sv` is a survey object created from `namcs2019sv_df` using `[survey::svydesign()]`.

Source

- SAS data: https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NAMCS/sas/namcs2019_sas.zip
- Survey design variables: https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NAMCS/sas/readme2019-sas.txt
- SAS formats: https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NAMCS/sas/nam19for.txt
- Documentation: https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NAMCS/doc2019-508.pdf
- National Summary Tables: https://www.cdc.gov/nchs/data/ahcd/namcs_summary/2019-namcs-web-tables-50
  pdf

`print.surveytable_table`

Print surveytable tables

Description

Print surveytable tables

Usage

```
## S3 method for class 'surveytable_table'
print(x, ...)

## S3 method for class 'surveytable_list'
print(x, ...)
```

Arguments

`x`	an object of class `surveytable_table` or `surveytable_list`.
`...`	ignored

Value

x invisibly.

Examples

```
set_survey(namcs2019sv)
table1 = tab("AGER")
print(table1)
table_many = tab("MDDO", "SPECCAT", "MSA")
print(table_many)
```

set_count_1k *Rounding counts*

Description

Determines how counts should be rounded.

Usage

```
set_count_1k()

set_count_int()
```

Details

- `set_count_1k()`: round counts to the nearest 1,000.
- `set_count_int()`: round counts to the nearest integer.

Value

(Nothing.)

See Also

Other options: `set_output()`, `show_options()`, `surveytable-options`

Examples

```
set_survey(namcs2019sv)
set_count_int()
total()

set_count_1k()
total()
```

`set_output` *Set output defaults*

Description

`show_output()` shows the current defaults.

Usage

```
set_output(csv = NULL, max_levels = NULL)

show_output()
```

Arguments

`csv`	name of a CSV file or "" to turn off CSV output
`max_levels`	a categorical variable can have at most this many levels. Used to avoid printing huge tables.

Value

(Nothing.)

See Also

Other options: `set_count_1k()`, `show_options()`, `surveytable-options`

Examples

```
tmp_file = tempfile(fileext = ".csv")
suppressMessages( set_output(csv = tmp_file) )
set_output(csv = "") # Turn off CSV output
```

`set_survey` *Specify the survey to analyze*

Description

You need to specify a survey before the other functions, such as `tab()`, will work.

Usage

```
set_survey(design)
```

Arguments

`design`	a survey object (`survey.design` or `svyrep.design`)

Details

Optionally, the survey can have an attribute called `label`, which is the long name of the survey.

Optionally, each variable in the survey can have an attribute called `label`, which is the variable's long name.

Value

Info about the survey.

Examples

```
set_survey(namcs2019sv)
```

---

show_options *Show package options*

Description

Show package options

Usage

```
show_options(sw = "surveytable")
```

Arguments

`sw`	starting characters

Value

List of options and their values.

See Also

Other options: `set_count_1k()`, `set_output()`, `surveytable-options`

Examples

```
show_options()
```

`surveytable-options` *Package options*

Description

Run `show_options()` to see available options.

See Also

Other options: `set_count_1k()`, `set_output()`, `show_options()`

`survey_subset` *Subset a survey, while preserving variable labels*

Description

Subset a survey, while preserving variable labels

Usage

```
survey_subset(design, subset, label)
```

Arguments

`design`	a survey object
`subset`	an expression specifying the sub-population
`label`	survey label of the newly created survey object

Value

a new survey object

Examples

```
children = survey_subset(namcs2019sv, AGE < 18, "Children < 18")
set_survey(children)
tab("AGER")
```

svyciprop_adjusted *Confidence intervals for proportions, adjusted for degrees of freedom*

Description

A version of `survey::svyciprop()` that adjusts for the degrees of freedom when `method = "beta"`.

Usage

```
svyciprop_adjusted(
  formula,
  design,
  method = c("logit", "likelihood", "asin", "beta", "mean", "xlogit"),
  level = 0.95,
  df_method,
  ...
)
```

Arguments

`formula`	see `survey::svyciprop()`.
`design`	see `survey::svyciprop()`.
`method`	see `survey::svyciprop()`.
`level`	see `survey::svyciprop()`.
`df_method`	how `df` should be calculated: "default" or "NHIS".
`...`	see `survey::svyciprop()`. `df_method`: for "default", `df = degf(design)`; for "NHIS", `df = nrow(design) - 1`.

Details

Written by Makram Talih in 2019.

To use this function in tabulations, type: `options(surveytable.adjust_svyciprop = TRUE)`.

Value

The point estimate of the proportion, with the confidence interval as an attribute.

Examples

```
set_survey(namcs2019sv)
options(surveytable.adjust_svyciprop = TRUE)
tab("AGER")
options(surveytable.adjust_svyciprop = FALSE)
tab("AGER")
```

tab *Tabulate variables*

Description

Tabulate categorical (factor), logical, or numeric variables.

Usage

```
tab(
  ...,
  test = FALSE,
  alpha = 0.05,
  drop_na = getOption("surveytable.drop_na"),
  max_levels = getOption("surveytable.max_levels"),
  csv = getOption("surveytable.csv")
)
```

Arguments

`...`	names of variables (in quotes)
`test`	perform hypothesis tests?
`alpha`	significance level for tests
`drop_na`	drop missing values (`NA`)? Categorical variables only.
`max_levels`	a categorical variable can have at most this many levels. Used to avoid printing huge tables.
`csv`	name of a CSV file

Details

For categorical and logical variables, presents the estimated counts, their standard errors (SEs) and confidence intervals (CIs), percentages, and their SEs and CIs. Checks the presentation guidelines for counts and percentages and flags estimates if, according to the guidelines, they should be suppressed, footnoted, or reviewed by an analyst.

For numeric variables, presents the percentage of observations with known values, the mean of known values, the standard error of the mean (SEM), and the standard deviation (SD).

CIs are calculated at the 95% confidence level. CIs for count estimates are the log Student's t CIs, with adaptations for complex surveys. CIs for percentage estimates are the Korn and Graubard CIs.

Value

A list of tables or a single table.

See Also

Other tables: `tab_cross()`, `tab_rate()`, `tab_subset_rate()`, `total_rate()`, `total()`

Examples

```
set_survey(namcs2019sv)
tab("AGER")
tab("MDDO", "SPECCAT", "MSA")

# Numeric variables
tab("NUMMED")

# Hypothesis testing with categorical variables
tab("AGER", test = TRUE)
```

---

tab_cross *Tabulate subsets or interactions*

Description

Create subsets of the survey using one variable, and tabulate another variable within each of the subsets. Interact two variables and tabulate.

Usage

```
tab_cross(
  vr,
  vrby,
  max_levels = getOption("surveytable.max_levels"),
  csv = getOption("surveytable.csv")
)

tab_subset(
  vr,
  vrby,
  lvls = c(),
  test = FALSE,
  alpha = 0.05,
  drop_na = getOption("surveytable.drop_na"),
  max_levels = getOption("surveytable.max_levels"),
  csv = getOption("surveytable.csv")
)
```

Arguments

`vr`	variable to tabulate
`vrby`	use this variable to subset the survey
`max_levels`	a categorical variable can have at most this many levels. Used to avoid printing huge tables.
`csv`	name of a CSV file

`lvls`	(optional) only show these levels of `vrby`
`test`	perform hypothesis tests?
`alpha`	significance level for tests
`drop_na`	drop missing values (`NA`)? Categorical variables only.

Details

`tab_subset` creates subsets using the levels of `vrby`, and tabulates `vr` in each subset. Optionally, only use the `lvls` levels of `vrby`. `vr` can be categorical (factor), logical, or numeric.

`tab_cross` crosses or interacts `vr` and `vrby` and tabulates the new variable. Tables created using `tab_subset` and `tab_cross` have the same counts but different percentages. With `tab_subset`, percentages within each subset add up to 100%. With `tab_cross`, percentages across the entire population add up to 100%. Also see `var_cross()`.

`test = TRUE` performs a test of association between the two variables. Also performs t-tests for all possible pairs of levels of `vr` and `vrby`.

Value

A list of tables or a single table.

See Also

Other tables: `tab_rate()`, `tab_subset_rate()`, `tab()`, `total_rate()`, `total()`

Examples

```
set_survey(namcs2019sv)

# For each SEX, tabulate AGER
tab_subset("AGER", "SEX")

# Same counts as tab_subset(), but different percentages.
tab_cross("AGER", "SEX")

# Numeric variables
tab_subset("NUMMED", "AGER")

# Hypothesis testing
tab_subset("NUMMED", "AGER", test = TRUE)
```

tab_rate *Calculate rates*

Description

Calculate the rates for categorical (factor) or logical variables.

Usage

```
tab_rate(
  vr,
  pop,
  per = getOption(”surveytable.rate_per”),
  drop_na = getOption(”surveytable.drop_na”),
  max_levels = getOption(”surveytable.max_levels”),
  csv = getOption(”surveytable.csv”)
)
```

Arguments

`vr` variable to tabulate

`pop` either a single number or a `data.frame` with columns named `Level` and `Population`. `Level` must exactly match the levels of `vr`. `Population` is the population for that level of `vr`.

`per` calculate rate per this many items in the population

`drop_na` drop missing values (`NA`)?

`max_levels` a categorical variable can have at most this many levels. Used to avoid printing huge tables.

`csv` name of a CSV file

Value

A list of tables or a single table.

See Also

Other tables: `tab_cross()`, `tab_subset_rate()`, `tab()`, `total_rate()`, `total()`

Examples

```
set_survey(namcs2019sv)
# pop is a data frame
tab_rate("MSA", uspop2019$MSA)

# pop is a single number
tab_rate("MDDO”, uspop2019$total)
```

`tab_subset_rate` *Calculate rates for subsets*

Description

Create subsets of the survey using one variable, and tabulate the rates of another variable within each of the subsets.

Usage

```
tab_subset_rate(
  vr,
  vrby,
  pop,
  lvls = c(),
  per = getOption("surveytable.rate_per"),
  drop_na = getOption("surveytable.drop_na"),
  max_levels = getOption("surveytable.max_levels"),
  csv = getOption("surveytable.csv")
)
```

Arguments

`vr` variable to tabulate

`vrby` use this variable to subset the survey

`pop` a `data.frame` with columns named `Level`, `Subset`, and `Population`. `Level` must exactly match the levels of `vr`. `Subset` must exactly match the levels of `vrby`. `Population` is the population for that level of `vr` and `vrby`.

`lvls` (optional) only show these levels of `vrby`

`per` calculate rate per this many items in the population

`drop_na` drop missing values (`NA`)?

`max_levels` a categorical variable can have at most this many levels. Used to avoid printing huge tables.

`csv` name of a CSV file

Value

A list of tables or a single table.

See Also

Other tables: `tab_cross()`, `tab_rate()`, `tab()`, `total_rate()`, `total()`

Examples

```
set_survey(namcs2019sv)
tab_subset_rate("AGER", "SEX", uspop2019$`AGER x SEX`)
```

total *Total count*

Description

Total count

Usage

```
total(csv = getOption("surveytable.csv"))
```

Arguments

csv	name of a CSV file

Value

A table

See Also

Other tables: tab_cross(), tab_rate(), tab_subset_rate(), tab(), total_rate()

Examples

```
set_survey(namcs2019sv)
total()
```

total_rate *Overall rate*

Description

Overall rate

Usage

```
total_rate(
  pop,
  per = getOption("surveytable.rate_per"),
  csv = getOption("surveytable.csv")
)
```

Arguments

`pop`	population
`per`	calculate rate per this many items in the population
`csv`	name of a CSV file

Value

A table

See Also

Other tables: `tab_cross()`, `tab_rate()`, `tab_subset_rate()`, `tab()`, `total()`

Examples

```
set_survey(namcs2019sv)
total_rate(uspop2019$total)
```

`uspop2019` *US Population in 2019*

Description

Population estimates of the civilian non-institutional population of the United States as of July 1, 2019. Used for calculating rates. For usage examples, see the `*_rate` functions.

Usage

```
uspop2019
```

Format

An object of class `list` of length 7.

var_any *Is any variable true?*

Description

Create a new variable which is true if any of the variables in a list of variables are true.

Usage

```
var_any(newvr, vrs)
```

Arguments

newvr	name of the new variable to be created
vrs	vector of logical variables

Value

Survey object

See Also

Other variables: var_case(), var_collapse(), var_copy(), var_cross(), var_cut()

Examples

```
set_survey(namcs2019sv)
var_any("Imaging services"
, c("ANYIMAGE", "BONEDENS", "CATSCAN", "ECHOCARD", "OTHULTRA"
, "MAMMO", "MRI", "XRAY", "OTHIMAGE"))
tab("Imaging services")
```

var_case *Convert factor to logical*

Description

Convert factor to logical

Usage

```
var_case(newvr, vr, cases)
```

Arguments

`newvr`	name of the new logical variable to be created
`vr`	factor variable
`cases`	one or more levels of `vr` that are converted to `TRUE`. All other levels are converted to `FALSE`.

Value

Survey object

See Also

Other variables: `var_any()`, `var_collapse()`, `var_copy()`, `var_cross()`, `var_cut()`

Examples

```
set_survey(namcs2019sv)
var_case("Preventive care visits", "MAJOR", "Preventive care")
tab("Preventive care visits")
var_case("Surgery-related visits"
, "MAJOR"
, c("Pre-surgery", "Post-surgery"))
tab("Surgery-related visits")
```

`var_collapse` *Collapse factor levels*

Description

Collapse two or more levels of a factor variable into a single level.

Usage

```
var_collapse(vr, newlevel, oldlevels)
```

Arguments

`vr`	factor variable
`newlevel`	name of the new level
`oldlevels`	vector of old levels

Value

Survey object

See Also

Other variables: `var_any()`, `var_case()`, `var_copy()`, `var_cross()`, `var_cut()`

Examples

```
set_survey(namcs2019sv)
tab("PRIMCARE")
var_collapse("PRIMCARE", "Unknown if PCP", c("Blank", "Unknown"))
tab("PRIMCARE")
```

var_copy *Copy a variable*

Description

Create a new variable that is a copy of another variable. You can modify the copy, while the original remains unchanged. See examples.

Usage

```
var_copy(newvr, vr)
```

Arguments

newvr	name of the new variable to be created
vr	variable

Value

Survey object

See Also

Other variables: var_any(), var_case(), var_collapse(), var_cross(), var_cut()

Examples

```
set_survey(namcs2019sv)
var_copy("Age group", "AGER")
var_collapse("Age group", "65+", c("65-74 years", "75 years and over"))
var_collapse("Age group", "25-64", c("25-44 years", "45-64 years"))
tab("AGER", "Age group")
```

var_cross *Cross or interact two variables*

Description

Create a new variable which is an interaction of two other variables. Also see `tab_cross()`.

Usage

```
var_cross(newvr, vr, vrby)
```

Arguments

`newvr`	name of the new variable to be created
`vr`	first variable
`vrby`	second variable

Value

Survey object

See Also

Other variables: `var_any()`, `var_case()`, `var_collapse()`, `var_copy()`, `var_cut()`

Examples

```
set_survey(namcs2019sv)
var_cross("Age x Sex", "AGER", "SEX")
tab("Age x Sex")
```

var_cut *Convert numeric to factor*

Description

Create a new categorical variable based on a numeric variable.

Usage

```
var_cut(newvr, vr, breaks, labels)
```

Arguments

newvr	name of the new factor variable to be created
vr	numeric variable
breaks	see cut()
labels	see cut()

Value

Survey object

See Also

Other variables: var_any(), var_case(), var_collapse(), var_copy(), var_cross()

Examples

```
set_survey(namcs2019sv)
var_cut("Age group"
, "AGE"
, c(-Inf, 0, 4, 14, 64, Inf)
, c("Under 1", "1-4", "5-14", "15-64", "65 and over"))
tab("Age group")
```

---

var_list *List variables in a survey.*

Description

List variables in a survey.

Usage

```
var_list(sw = "", all = FALSE, csv = getOption("surveytable.csv"))
```

Arguments

sw	starting characters in variable name (case insensitive)
all	print all variables?
csv	name of a CSV file

Value

A table

Examples

```
set_survey(namcs2019sv)
var_list("age")
```

Index