Package 'nonprobsampling'

July 7, 2026

Type Package

Title Inference for Nonprobability Samples Using Multiple Reference Surveys

Version 0.1.0

Description Provides pseudo-weighted estimates of means and prevalences for finite population inference from nonprobability samples using auxiliary information from one or multiple probability reference surveys. The package supports estimation with multiple reference surveys, allowing auxiliary information to be combined when no single survey contains all variables relevant to participation. Optional cumulative precalibration can be applied to align weighted totals of shared variables across surveys. Methods are based on the generalized estimating equations framework of Landsman et al. (2026) <doi:10.1002/sim.70403> for correcting participation bias. For a single reference survey, the package implements the raking ratio calibration method and includes the adjusted logistic propensity (ALP) method of Wang, Valliant, and Li (2021) <doi:10.1002/sim.9122>, as well as the Chen-Li-Wu (CLW) method of Chen, Li, and Wu (2020) <doi:10.1080/01621459.2019.1677241>. Analytic variance estimation uses Taylor linearization and accounts for complex sampling designs in the reference surveys via integration with the 'survey' package.

Depends R (>= 4.0.0)

Imports stats, survey, nleqslv, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Language en-US

URL https://github.com/Jiakun0611/nonprobsampling,
https://jiakun0611.github.io/nonprobsampling/

BugReports https://github.com/Jiakun0611/nonprobsampling/issues

Config/testthat/edition 3

NeedsCompilation no

Author Jiakun Lin [aut, cre],
Victoria Landsman [aut],
Aya A. Mitani [aut]

Maintainer Jiakun Lin <jiak.lin@alumni.utoronto.ca>

Repository CRAN

Date/Publication 2026-07-07 10:00:02 UTC

Contents

`est_pw` *Estimate Pseudo-Weights for Nonprobability Samples*

Description

`est_pw()` estimates pseudo-weights for a nonprobability sample using one reference survey or multiple reference surveys. The function specifies the participation model, handles missing values in the participation model variables, solves the estimating equations, and stores the quantities needed for downstream point and variance estimation.

Users should harmonize variable names and coding before calling `est_pw()`. Variables used in the participation model must have consistent names and compatible definitions across the nonprobability sample and the reference survey data used for estimation.

With one reference survey, the available methods include the raking ratio calibration method described in Landsman et al. (2026), the adjusted logistic propensity weighting (ALP) method proposed by Wang, Valliant, and Li (2021), and the CLW method proposed by Chen, Li, and Wu (2020). With multiple reference surveys, pseudo-weights are estimated using the multi-reference calibration method proposed by Landsman et al. (2026).

The returned object is designed to be passed to `pwmean`.

Usage

```
est_pw(
  data,
  sp_order = c("size", "given"),
  precali = TRUE,
  p_formula = NULL,
  method = NULL,
  na.action = stats::na.omit,
  sc_wname = "pseudo_wts",
  control = pw_solver_control(),
  verbose = FALSE
)
```

Arguments

`data`	A list of input data objects of the form `list(sc, sp1_design, sp2_design, ...)`. The first element must be a data frame corresponding to the nonprobability sample. Each remaining element must be a valid survey design object corresponding to a reference probability survey, such as an object created by `svydesign` or `svrepdesign`.
`sp_order`	Character string controlling the order of reference surveys when multiple reference surveys are used. Supported values are `"size"` and `"given"`. `"size"` orders reference surveys by sample size, from largest to smallest. `"given"` uses the user-specified order of the reference surveys in `data`. Default is `"size"`. With one reference survey, this argument is ignored; a warning is issued if a non-default value is supplied.
`precali`	Logical. Used only with multiple reference surveys. If `TRUE`, cumulative precalibration is applied before the main multi-reference estimation step; see the **Multi-reference precalibration** section for details. Default is `TRUE`. With one reference survey, this argument is ignored; a warning is issued if `FALSE` is supplied.
`p_formula`	Optional participation model formula. Must always be one-sided (no response variable on the left-hand side). A two-sided formula such as `y ~ x` will raise an error. With one reference survey, supply a single one-sided formula, for example `~ age + sex + income`. With multiple reference surveys, supply a list of one-sided formulas with one formula per reference survey, for example `list(~ age + sex,`

`~ age + income`). If `NULL`, a default formula is constructed automatically from variables shared across the data sources used for estimation. Since shared variables are identified by name, their names should be harmonized across data sources before estimation.

`method` Character string specifying the pseudo-weighting method, or `NULL` (default). If `NULL`, `"calibration"` is used when `data` contains one reference survey, and `"multi"` is used when `data` contains more than one reference survey.

To override the default, supply one of the following values. For a one-reference method: `"alp"`, `"clw"`, or `"calibration"` (or `"cali"`). For the multi-reference method: `"multi"`.

The argument is case-insensitive, so inputs such as `"ALP"`, `"Clw"`, or `"CALI"` are also accepted.

`na.action` Function specifying how missing values should be handled for variables used in the participation model. Common choices include `stats::na.omit`, `stats::na.exclude`, and `stats::na.fail`. Default is `stats::na.omit`.

`sc_wname` Character string giving the name of the pseudo-weight column added to the returned nonprobability sample. Default is `"pseudo_wts"`. An error is raised at input validation if this name already exists as a column in `sc`.

`control` A solver control object created by `pw_solver_control`. This object stores numerical settings for solving estimating equations, including the solver, convergence tolerance, maximum number of iterations, tracing behavior, and other options.

`verbose` Logical. If `TRUE`, progress messages and diagnostics are printed during pseudo-weight estimation. Default is `FALSE`. Must be a single `TRUE` or `FALSE`; an error is raised otherwise.

Details

`est_pw()` performs pseudo-weight estimation for the nonprobability sample and stores the method-specific internal objects needed later by `pwmean`. It does not require an outcome variable.

The input `data` must be provided as a list, where the first element is the nonprobability sample and the remaining elements are reference survey design objects. Reference survey designs can be created with `svydesign` for standard complex survey designs or `svrepdesign` for surveys with replicate weights. These objects preserve the sampling structure needed for design-consistent variance estimation.

Variable harmonization. Variables are matched by name, not by meaning. Before applying `est_pw()`, shared variables must be harmonized across the nonprobability sample and reference survey data. For example, if a categorical variable is named `agecat` in the nonprobability sample and `age_group` in the reference survey, the user should rename one of the variables before estimation.

Categorical variables should be encoded as factors with compatible category definitions and identical levels in the same order. Even when categories are substantively equivalent, mismatched factor levels may cause `est_pw()` to return an error. Continuous variables included in the participation model should also be measured on comparable scales across datasets.

Internally, `est_pw()` performs the following steps:

1. **Input validation**
   Validates the structure and required components of the input data.
2. **Reference survey detection**
   Determines whether the input contains a single reference survey or multiple reference surveys.
3. **Method selection**
   Selects the pseudo-weighting method based on the specified argument(s).
4. **Participation model specification**
   Constructs a default participation model formula when `p_formula = NULL`.
5. **Missing data handling**
   Applies missing-data handling procedures to variables used in the participation model.
6. **Model matrix construction**
   Generates model matrices from the participation model variables.
7. **Pseudo-weight estimation**
   Estimates pseudo-weights using the selected method.
8. **Output augmentation**
   Appends the estimated pseudo-weights as a new column to the nonprobability sample.
9. **Metadata storage**
   Stores information related to missing-data handling and other internal objects for later use or diagnostics.

Value

An object of class `"pw_fit"`. This is a list containing user-facing outputs and internal objects required by `pwmean`.

Important components include:

`sc_updated` A data frame containing the nonprobability sample with an added pseudo-weight column named by `sc_wname`.

`pseudo_weights` The estimated pseudo-weight vector. With `stats::na.omit`, the vector contains only observations retained for pseudo-weight estimation. With `stats::na.exclude`, excluded observations receive `NA` and the vector has length `nrow(sc)`.

`coefficients` Estimated coefficients for the participation model variables.

`solver_diagnostics` A list of solver diagnostics: `solver` (solver name), `termcd` (termination code), `message` (solver message), `iter` (number of iterations), and `fmax` (maximum absolute value of the final estimating equations at convergence).

`method` The pseudo-weighting method used by the function.

`internal` A list of internal objects needed for downstream estimation.

`na_summary` An object of class `"pw_na_summary"` summarizing the number of rows excluded from the nonprobability sample and each reference survey due to missing participation model variables. `NULL` if no rows were excluded.

`call` The matched function call.

One-reference method and multi-reference method

If `data` contains one reference survey design object, `est_pw()` fits a one-reference method. If `data` contains more than one reference survey design objects, `est_pw()` fits the multi-reference calibration method. In both settings, the auxiliary variables used for pseudo-weight estimation should be harmonized across all data sources before calling `est_pw()`.

Multi-reference precalibration

When `precali = TRUE`, cumulative precalibration is performed before the main multi-reference calibration step. For overlapping auxiliary variables, this procedure calibrates the survey weights of a reference survey so that its weighted totals of the overlapping variables and its sum of weights match the corresponding totals from the preceding reference survey in the cumulative order. If there are no overlapping auxiliary variables, cumulative precalibration is applied only to the sum of weights.

The order of the reference surveys is controlled by `sp_order`. If `sp_order = "size"`, reference surveys are ordered by sample size, from largest to smallest. If `sp_order = "given"`, the user-specified order of the reference surveys is used.

Cumulative precalibration is based only on overlapping variables that are specified in `p_formula`, rather than on all overlapping variables in the reference surveys. This choice avoids excluding observations because of missing values in variables that are not used for pseudo-weight estimation.

Missing data handling

Missing values are handled only for variables used in the participation model. The selected `na.action` is recorded in the returned object, together with the row indices of the nonprobability sample observations retained for pseudo-weight estimation.

With `stats::na.omit`, rows with missing participation model variables are removed from `sc_updated`. With `stats::na.exclude`, the original rows are retained in `sc_updated`, but excluded rows receive `NA` in the pseudo-weight column. This can be useful when users want to preserve row alignment with the original nonprobability sample for later imputation or merging.

Numerical control

Numerical settings are supplied through the `control` argument, which should be created by `pw_solver_control`. This object controls solver choice, convergence tolerance, maximum iterations, tracing, and optional solver-specific arguments.

The top-level `ftol`, `xtol`, and `maxit` values in `pw_solver_control` are the package-level convergence controls used by pseudo-weight estimation stages. When the selected solver is `"nleqslv"`, additional arguments can be passed through `nleqslv_control`. These are forwarded to `nleqslv::nleqslv()`.

References

Chen, Y., Li, P., and Wu, C. (2020). Doubly robust inference with nonprobability survey samples. *Journal of the American Statistical Association*, 115(532), 2011–2021. doi:10.1080/01621459.2019.1677241

Wang, L., Valliant, R., and Li, Y. (2021). Adjusted logistic propensity weighting methods for population inference using nonprobability volunteer-based epidemiologic cohorts. *Statistics in Medicine*, 40(24), 5237–5250. doi:10.1002/sim.9122

Landsman, V., Wang, L., Carrillo-Garcia, I., Mitani, A. A., Smith, P. M., Graubard, B. I., Bui, T., and Carnide, N. (2026). Correction for Participation Bias in Nonprobability Samples Using Multiple Reference Surveys. *Statistics in Medicine*, 45(3–5). doi:10.1002/sim.70403

See Also

pw_solver_control, pwmean

Examples

```
data(sc)
data(sp1)
data(sp2)

## One-reference example

ref1_design <- survey::svydesign(
  ids     = ~psu_sp1,
  strata  = ~strata_sp1,
  weights = ~wts_sp1,
  data    = sp1,
  nest    = TRUE
)

fit1 <- est_pw(
  data      = list(sc, ref1_design),
  p_formula = ~ agecat + race + education + comorbidity + BMI + diabetes,
  method    = "calibration",
  control   = pw_solver_control(ftol = 1e-6)
)

print(fit1)

summary(fit1)

## Multi-reference example

ref2_design <- survey::svydesign(
  ids     = ~psu_sp2,
  strata  = ~strata_sp2,
  weights = ~wts_sp2,
  data    = sp2,
  nest    = TRUE
)

fit2 <- est_pw(
  data = list(sc, ref1_design, ref2_design),
  p_formula = list(
    ~ agecat + race + education + psa_level + pros_enlarged + comorbidity,
    ~ agecat + race + BMI + diabetes + comorbidity
  ),
  sp_order = "size",
```

```
  precali = TRUE,
  control = pw_solver_control(ftol = 1e-6)
)

print(fit2)

summary(fit2)
```

`na.action.pwmean` *Extract NA action from a pwmean object*

Description

Returns the `na.action` component recorded during estimation, mimicking `na.action` behavior for fitted model objects.

Usage

```
## S3 method for class 'pwmean'
na.action(object, ...)
```

Arguments

`object`	An object of class `"pwmean"` returned by `pwmean`.
`...`	Additional arguments (not used).

Value

The `na.action` object recorded by `pwmean`: an integer vector of rows omitted because of missing outcome or domain values (of class `"omit"` or `"exclude"`), or `NULL` if no rows were omitted.

`na.action.pw_fit` *Extract NA action from a pw_fit object*

Description

Returns the `na.action` component recorded during the build step.

Usage

```
## S3 method for class 'pw_fit'
na.action(object, ...)
```

Arguments

object	An object of class "pw_fit" returned by est_pw.
...	Additional arguments (not used).

Value

The na.action object recorded by est_pw during the build step: an integer vector of the nonprobability-sample rows omitted because of missing participation model variables (of class "omit" or "exclude"), or NULL if no rows were omitted.

---

print.pwmean *Print method for pwmean objects*

Description

Displays the pseudo-weighted mean estimate and its uncertainty. For factor-like domain variables, prints one row per domain level.

Usage

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

Arguments

x	An object of class "pwmean", returned by pwmean.
...	Additional arguments, currently unused.

Value

Invisibly returns x.

---

print.pwmean_factor *Print method for pwmean objects with categorical outcomes*

Description

Displays pseudo-weighted prevalence estimates and their uncertainty.

Usage

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

Arguments

x	An object of class "pwmean_factor", returned by pwmean when y is a factor.
...	Additional arguments, currently unused.

Value

Invisibly returns x.

---

print.pw_fit *Print method for pw_fit objects*

Description

Compact one-screen overview of a fitted pseudo-weight object: the call, the pseudo-weighting method, the participation model size, solver convergence, and a summary of the estimated pseudo-weights. For the full coefficient table and detailed solver diagnostics, use summary.pw_fit.

Usage

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

Arguments

x	An object of class "pw_fit", returned by est_pw.
...	Additional arguments, currently unused.

Value

Invisibly returns x.

---

print.pw_na_summary *Print method for pw_na_summary*

Description

Prints a formatted table showing the original row count, rows used, and rows excluded due to missing participation model variables, for each dataset (‘sc‘ and each reference survey).

Usage

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

Arguments

x	A ‘pw_na_summary‘ object returned by ‘.report_na_exclusions()‘.
...	Further arguments passed to or from other methods (unused).

Value

Invisibly returns ‘x‘.

---

pwmean *Estimate Pseudo-Weighted Means, Prevalences, and Standard Errors*

Description

Computes pseudo-weighted means and standard errors using a fitted pseudo-weight object of class `"pw_fit"` returned by `est_pw`. The function applies second-layer missing-data handling for the outcome and optional domain variable, and then estimates overall or domain-specific means or prevalences using the pseudo-weighting method stored in `object`.

Usage

```
pwmean(object, y, zcol = NULL, na.action = stats::na.omit)
```

Arguments

`object`	An object of class `"pw_fit"` returned by `est_pw`.
`y`	A character string specifying the name of the outcome variable in the nonprobability sample stored in `object`. The outcome must be numeric for mean estimation, including binary 0/1 outcomes for prevalence estimation, or a factor for category prevalence estimation.
`zcol`	Optional character string giving the name of a categorical domain variable in the nonprobability sample stored in the `object`. If `NULL`, the overall mean is estimated. If supplied, estimates are computed within domains defined by this variable. The following column types are supported: `logical` (must contain both `TRUE` and `FALSE`); `numeric` or `integer` containing only `0` and `1` after removing missing values; `character` (empty strings are treated as missing values); and `factor` (unused levels are dropped).
`na.action`	Function specifying how missing values in `y` and `zcol` should be handled. Default is `stats::na.omit`.

Details

Missing data handling (layer 2). After pseudo-weights are constructed by `est_pw()`, estimation of the mean requires complete cases for the outcome `y` and, if supplied, the domain variable `zcol`. The argument `na.action` controls how these missing values are handled at the outcome-estimation step.

Input `object`**.** The `object` argument should be an object of class `"pw_fit"` returned by `est_pw`. It stores the estimated pseudo-weights, participation model information, and design-based quantities required for point and variance estimation.

Categorical outcomes. When `y` is a categorical variable (defined as a factor in R), `pwmean()` estimates the prevalence (proportion) of each category. To do so, each category is internally converted into a 0/1 indicator variable, and the pseudo-weighted mean estimator is then computed for each indicator.

Value

An object of class `"pwmean"` containing unweighted and pseudo-weighted estimates, standard errors, and confidence intervals. For categorical outcomes, the estimate columns contain category prevalences.

`method` Character. The pseudo-weighting method used.

`estimates` A data frame containing the unweighted and pseudo-weighted estimates.

For numeric outcomes, the first column is `domain`. If `zcol = NULL`, `domain` is `"Overall"`. If `zcol` is a `logical` variable or a `numeric/integer` variable containing only `0` and `1`, there is one row with `domain` labeled `"<zcol> = 1"`. If `zcol` is a `factor` or `character` variable, there is one row per `zcol` level, with `domain` labeled `"<zcol> = <level>"`.

For categorical outcomes, the first two columns are `category` and `domain`. `category` identifies the outcome level as `"<y> = <level>"`. If `zcol = NULL`, `domain` is `"Overall"` for each outcome level. If `zcol` is supplied, the rows are formed by each outcome category within each domain, and `domain` follows the same labels described above for `zcol`.

The columns are:

`category` Category label for categorical outcomes only.

`domain` Domain label.

`unweighted_mean`**,** `unweighted_se` Unweighted mean of `y` and its standard error.

`unweighted_lower`**,** `unweighted_upper` Bounds of the 95% confidence interval for the unweighted mean, based on the normal approximation.

`adjusted_mean`**,** `adjusted_se` Pseudo-weighted mean of `y` and its standard error.

`adjusted_lower`**,** `adjusted_upper` Bounds of the 95% confidence interval for the pseudo-weighted mean, based on the normal approximation.

`na.action` Integer vector of row indices omitted at the outcome-estimation step, with class `"omit"` or `"exclude"` matching the `na.action` argument, or `NULL` if no observations were omitted. The indices refer to the nonprobability sample available to `pwmean()` after missing-data handling in `est_pw()`.

`call` The matched function call.

See Also

`est_pw`, `summary.pwmean`, `print.pwmean`

Examples

```
data(sc)
data(sp1)

ref1_design <- survey::svydesign(
  ids     = ~psu_sp1,
  strata  = ~strata_sp1,
  weights = ~wts_sp1,
  data    = sp1,
  nest    = TRUE
)

fit <- est_pw(
  data      = list(sc, ref1_design),
  p_formula = ~ agecat + race + education + comorbidity + BMI + diabetes,
  method    = "calibration",
  control   = pw_solver_control(ftol=1e-6)
)

out <- pwmean(fit, y = "psa_level", zcol = "BMI")

print(out)

summary(out)
```

`pw_solver_control` *Control Solver Settings for Pseudo-Weight Estimation*

Description

`pw_solver_control()` creates a solver control object used by `est_pw` to manage numerical settings for pseudo-weight estimation.

Usage

```
pw_solver_control(
  solver = "nleqslv",
  maxit = NULL,
  trace = FALSE,
  method = c("Newton", "Broyden"),
  global = c("dbldog", "cline", "pwldog", "qline", "gline", "hook", "none"),
  xscalm = c("fixed", "auto"),
  ftol = 1e-08,
  xtol = 1e-08,
  nleqslv_control = list()
)
```

Arguments

solver	Character string specifying the numerical solver used for solving the estimating equations. Currently, only "nleqslv" is supported. Default is "nleqslv".
maxit	Positive finite numeric value passed to nleqslv::nleqslv() as the maximum number of solver iterations. The value is converted to an integer before being stored in the returned control object. Default is 150 when a global strategy is specified (i.e., global != "none"), and 20 when no global strategy is used (global = "none"), matching nleqslv's own defaults. Since the default global strategy is "dbldog", the effective default is 150 unless global = "none" is explicitly specified.
trace	Logical. If TRUE, tracing or solver progress information may be requested from the underlying numerical routine when supported. Default is FALSE.
method	Character string specifying the numerical method passed to nleqslv::nleqslv(). Supported values are "Newton" and "Broyden". Default is "Newton".
global	Character string specifying the global strategy passed to nleqslv::nleqslv(). Supported values are "dbldog", "cline", "pwldog", "qline", "gline", "hook", and "none". Default is "dbldog".
xscalm	Character string specifying the scaling method passed to nleqslv::nleqslv(). Supported values are "fixed" and "auto". Default is "fixed".
ftol	Positive finite numeric value passed to nleqslv::nleqslv() as the function-value convergence tolerance. This controls convergence based on the size of the estimating function. Default is 1e-8.
xtol	Positive finite numeric value passed to nleqslv::nleqslv() as the parameter-step convergence tolerance. This controls convergence based on changes in the parameter vector. Default is 1e-8.
nleqslv_control	A list of additional control options passed to nleqslv::nleqslv(). This can include less commonly used control options, such as btol, cndtol, sigma, and scalex. See nleqslv for details.

Details

The control object stores solver settings used by pseudo-weight estimation step. It is passed to est_pw through the control argument.

Currently, only solver = "nleqslv" is supported. The arguments method, global, xscalm, ftol, xtol, and maxit correspond to options used by nleqslv::nleqslv(). They are collected internally and passed to nleqslv::nleqslv() at the pseudo-weight estimation step.

The argument ftol is the function-value convergence tolerance. It controls convergence based on the size of the estimating function. The argument xtol is the parameter-step convergence tolerance. It controls convergence based on changes in the parameter vector. The argument maxit controls the maximum number of solver iterations.

Additional, less commonly used nleqslv control options can be supplied through nleqslv_control. To avoid ambiguity, do not supply ftol, xtol, maxit, or trace inside nleqslv_control; use the main arguments instead.

Value

A flat list containing all solver control settings for pseudo-weight estimation:

`solver` The selected numerical solver.

`method` The nleqslv numerical method.

`global` The nleqslv global strategy.

`xscalm` The nleqslv scaling method.

`ftol` The function-value convergence tolerance.

`xtol` The parameter-step convergence tolerance.

`maxit` The maximum number of solver iterations, stored as an integer. `150` if a global strategy is used; `20` if `global = "none"`. Since the default global strategy is `"dbldog"`, the effective default is `150` unless `global = "none"` is explicitly specified.

`trace` Logical value indicating whether tracing information is requested.

`nleqslv_control` A list of additional options passed to `nleqslv::nleqslv()`.

See Also

`est_pw`

Examples

```
## Default solver control settings
ctrl <- pw_solver_control()

## Custom nleqslv solver settings
ctrl <- pw_solver_control(
  maxit  = 20,
  trace  = FALSE,
  method = "Newton",
  global = "cline",
  xscalm = "auto",
  ftol   = 1e-8,
  xtol   = 1e-10
)

## Additional nleqslv control options
ctrl <- pw_solver_control(
  method = "Newton",
  global = "dbldog",
  nleqslv_control = list(
    btol = 1e-3
  )
)
```

sc *Nonprobability Sample (sc)*

Description

This dataset represents a synthetic nonprobability sample generated via Poisson sampling from a finite population constructed from the National Health and Nutrition Examination Survey (NHANES) cycles 1999–2010. It is intended to illustrate the pseudo-weighting methods implemented in the `nonprobsampling` package.

Usage

```
data(sc)
```

Format

A data frame with 2404 observations and 8 variables:

psa_level Outcome variable: serum prostate-specific antigen level (numeric)

BMI Body mass index category (factor with 4 levels: "Normal", "Overweight", "Obese", "Morbidly Obese")

race Race category (factor with 4 levels: 1 = White, 2 = Black, 3 = Hispanic, 4 = Other)

agecat Age category (factor with 4 levels: 1 = 55–59, 2 = 60–64, 3 = 65–69, 4 = 70+)

education Education level (factor with 5 levels: 1 = Less Than 8 Years, 2 = 8–11 Years, 3 = 12 Years Or Completed High School, 4 = College Graduate, 5 = Postgraduate)

pros_enlarged Prostate enlargement indicator (factor with 2 levels: 0 = No, 1 = Yes)

comorbidity General comorbidity indicator (factor with 2 levels: 0 = No, 1 = Yes)

diabetes Diabetes diagnosis indicator (factor with 2 levels: 0 = No, 1 = Yes)

Details

The dataset has 2,404 complete-case observations, with `psa_level` serving as the outcome variable. Auxiliary variables shared with the probability reference surveys `sp1` and `sp2` are used to construct pseudo-weights aimed at correcting for participation bias.

Source

Synthetic data generated by the package authors. The underlying finite population was constructed from the National Health and Nutrition Examination Survey (NHANES), 1999–2010 cycles, conducted by the U.S. National Center for Health Statistics (NCHS).

Examples

```
data(sc)
str(sc)
summary(sc)
```

sp1 *Probability Reference Sample 1 (sp1)*

Description

This dataset represents a probability sample derived from the 1999–2010 cycles of the National Health and Nutrition Examination Survey (NHANES). It is used as a probability reference survey to support the pseudo-weighting methods implemented in the `nonprobsampling` package.

Usage

```
data(sp1)
```

Format

A data frame with 3494 observations and 14 variables:

agecat Age category (factor with 4 levels: 1 = 55–59, 2 = 60–64, 3 = 65–69, 4 = 70+)

marital Marital status (factor with 4 levels: 1 = Married Or Living As Married, 2 = Widowed, 3 = Divorced or Separated, 4 = Never Married)

race Race category (factor with 4 levels: 1 = White, 2 = Black, 3 = Hispanic, 4 = Other)

education Education level (factor with 5 levels: 1 = Less Than 8 Years, 2 = 8–11 Years, 3 = 12 Years Or Completed High School, 4 = College Graduate, 5 = Postgraduate)

employment Employment status (factor with 2 levels: 0 = Not Working, 1 = Working)

smoking Smoking status (factor with 3 levels: 1 = Never Smoker, 2 = Former Smoker, 3 = Current Smoker)

comorbidity General comorbidity indicator (factor with 2 levels: 0 = No, 1 = Yes)

psa_level Serum prostate-specific antigen level (numeric)

BMI Body mass index category (factor with 4 levels: "Normal", "Overweight", "Obese", "Morbidly Obese")

diabetes Diabetes diagnosis indicator (factor with 2 levels: 0 = No, 1 = Yes)

pros_enlarged Prostate enlargement indicator (factor with 2 levels: 0 = No, 1 = Yes)

strata_sp1 Stratum identifier for complex survey design (numeric)

psu_sp1 Primary sampling unit identifier for complex survey design (numeric)

wts_sp1 10-year interview sampling weights (numeric)

Details

The dataset includes auxiliary variables shared with the nonprobability sample `sc`, enabling the construction of pseudo-weights to adjust for participation bias. Survey design variables and sampling weights are provided to support design-consistent estimation.

The `sp1` dataset contains the outcome variable `psa_level`, which is also observed in `sc`, allowing for the evaluation of pseudo-weighted estimators against estimates based on true sampling weights. It may also be incorporated into the participation model, potentially enhancing bias reduction when participation depends on the outcome.

Source

Derived from the National Health and Nutrition Examination Survey (NHANES), 1999–2010 cycles, conducted by the U.S. National Center for Health Statistics (NCHS).

Examples

```
data(sp1)
str(sp1)
summary(sp1)
```

`sp1_bootstrap` *Probability Reference Sample 1 with Bootstrap Replicate Weights (sp1_bootstrap)*

Description

A replicate-weight version of `sp1`, including the main survey weight and 500 bootstrap replicate weights. It is provided to illustrate design-based variance estimation with `svrepdesign`. The original primary sampling unit and stratum identifiers are not included.

Usage

```
data(sp1_bootstrap)
```

Format

A data frame with 3494 rows and 512 columns. The first 12 columns are the substantive survey variables from `sp1` (`agecat`, `marital`, `race`, `education`, `employment`, `smoking`, `comorbidity`, `psa_level`, `BMI`, `diabetes`, `pros_enlarged`, `wts_sp1`). The remaining 500 columns are bootstrap replicate weights named `bw1` through `bw500` (numeric).

Details

The bootstrap replicate weights were constructed from the original stratified cluster design of `sp1`, using the Rao-Wu rescaling bootstrap method. A total of `R = 500` replicates were produced with `seed = 2026`.

The variables `psu_sp1` and `strata_sp1` are not included in this dataset because they are not needed when using replicate weights for variance estimation. The main survey weight `wts_sp1` and the replicate weight columns `bw1`–`bw500` are sufficient for constructing a replicate-weight survey design object via `survey::svrepdesign()`.

Source

Derived from `sp1` (National Health and Nutrition Examination Survey, NHANES, 1999–2010 cycles), with bootstrap replicate weights added by the package authors using the Rao-Wu rescaling bootstrap.

Examples

```
data(sp1_bootstrap)

# Example: create replicate-weight survey design object
des_boot <- survey::svrepdesign(
  data    = sp1_bootstrap,
  weights = ~wts_sp1,
  repweights = "bw[0-9]+",
  type    = "bootstrap",
  combined.weights = FALSE
)

summary(des_boot)
```

---

sp2 *Probability Reference Sample 2 (sp2)*

Description

This dataset represents a probability survey derived from the 1997–2008 cycles of the National Health Interview Survey (NHIS). It is intended for use alongside `sc` and `sp1` to illustrate the multi-reference calibration method implemented in the `nonprobsampling` package.

Usage

```
data(sp2)
```

Format

A data frame with 35525 observations and 11 variables:

agecat Age category (factor with 4 levels: 1 = 55–59, 2 = 60–64, 3 = 65–69, 4 = 70+)

marital Marital status (factor with 4 levels: 1 = Married Or Living As Married, 2 = Widowed, 3 = Divorced or Separated, 4 = Never Married)

race Race category (factor with 4 levels: 1 = White, 2 = Black, 3 = Hispanic, 4 = Other)

employment Employment status (factor with 2 levels: 0 = Not Working, 1 = Working)

diabetes Diabetes diagnosis indicator (factor with 2 levels: 0 = No, 1 = Yes)

BMI Body mass index category (factor with 4 levels: "Normal", "Overweight", "Obese", "Morbidly Obese")

smoking Smoking status (factor with 3 levels: 1 = Never Smoker, 2 = Former Smoker, 3 = Current Smoker)

comorbidity General comorbidity indicator (factor with 2 levels: 0 = No, 1 = Yes)

wts_sp2 Sampling weights (numeric)

strata_sp2 Stratum identifier for complex survey design (numeric)

psu_sp2 Primary sampling unit identifier for complex survey design (numeric)

Details

The dataset includes auxiliary variables shared with the nonprobability sample `sc`, enabling the construction of pseudo-weights to adjust for participation bias. Survey design variables and sampling weights are provided to support design-consistent estimation.

Source

Derived from the National Health Interview Survey (NHIS), 1997–2008 cycles, conducted by the U.S. National Center for Health Statistics (NCHS).

Examples

```
data(sp2)
str(sp2)
summary(sp2)
```

`summary.pwmean` *Summary method for pwmean objects*

Description

Provides console output for objects of class `"pwmean"`, including unweighted and pseudo-weighted mean estimates, standard errors, confidence intervals, and optional domain-level summaries.

Usage

```
## S3 method for class 'pwmean'
summary(object, ...)
```

Arguments

`object`	An object of class `"pwmean"`, returned by `pwmean`.
`...`	Additional arguments, currently unused.

Value

Invisibly returns `object`.

summary.pwmean_factor *Summary method for pwmean objects with categorical outcomes*

Description

Provides console output for objects of class "pwmean_factor", including unweighted and pseudo-weighted prevalence estimates, standard errors, and confidence intervals.

Usage

```
## S3 method for class 'pwmean_factor'
summary(object, ...)
```

Arguments

object	An object of class "pwmean_factor", returned by pwmean when y is a factor.
...	Additional arguments, currently unused.

Value

Invisibly returns object.

summary.pw_fit *Summarize a Pseudo-Weight Fit*

Description

Summarize a Pseudo-Weight Fit

Usage

```
## S3 method for class 'pw_fit'
summary(object, ...)
```

Arguments

object	An object of class "pw_fit", returned by est_pw.
...	Additional arguments, currently unused.

Value

Invisibly returns object.

Index