Package 'rwetools'

September 9, 2026

Title Estimating Propensity Scores (PS), PS-Based Weights, and Effects

Version 0.5.0

Description Toolbox that provides a streamlined, end-to-end workflow for propensity score analysis in generating real-world evidence from real-world data. The package covers the full analytic pipeline - from estimating propensity scores via logistic regression, to calculating weights or creating a matched cohort, to generating publication-ready Table 1s with standardized mean differences and weighted balance diagnostics. It also estimates incidence rates, rate differences and rate ratios, hazard ratios, risks, risk ratios, and risk differences, including competing-risk methods and optionally stratified hazard models. Many functions can write formatted 'Excel' reports with method documentation, making results immediately shareable with collaborators and stakeholders. Methods are based on Rosenbaum and Rubin (1983) <doi:10.1093/biomet/70.1.41>, Austin (2011) <doi:10.1080/00273171.2011.568786>, and Desai et al. (2017) <doi:10.1097/EDE.0000000000000595>.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports stats, utils, parallel, survival, survey

Suggests openxlsx, ggplot2, MatchIt, testthat (>= 3.0.0)

URL https://github.com/hanseul0618/rwetools

Config/testthat/edition 3

NeedsCompilation no

Author Hanseul Cho [aut, cre],
Georg Hahn [aut],
Janinne Ortega-Montiel [aut],
Julie Paik [aut],
Elisabetta Patorno [aut]

Maintainer Hanseul Cho <hanseul0618@gmail.com>

Repository CRAN

Date/Publication 2026-09-09 16:00:02 UTC

Contents

`build_table1` *Build an Unweighted or Weighted Baseline-Characteristics Table*

Description

Summarizes continuous, binary, and categorical characteristics for the total cohort and two exposure arms. The output includes formatted counts or means, exposed-minus-reference crude differences, standardized mean differences (SMD), and missingness. SMDs are reported on the raw scale; for example, 0.1 rather than 10 percent.

Usage

```
build_table1(
  in_df,
  out_xlsxpath = NULL,
  exposure_var,
  exp_value = 1,
  ref_value = 0,
  use_weights = FALSE,
  weight_var = "psweight",
  cont_vars = NULL,
  cat_vars = NULL,
  binary_vars = NULL,
  drop_vars = NULL,
  drop_varpattern = NULL,
  Var_colname = "Variable",
  Vartype_colname = "Type",
  Total_colname = "Total",
  Exp_colname = "Exp",
```

```
  Ref_colname = "Ref",
  CrudeDiff_colname = "Crude_diff",
  StdDiff_colname = "Std_diff",
  MissingTotal_colname = "Missing_Total_N_Pct",
  MissingExp_colname = "Missing_Exp_N_Pct",
  MissingRef_colname = "Missing_Ref_N_Pct",
  ExistingTotal_colname = "Existing_Total_N_denom",
  ExistingExp_colname = NULL,
  ExistingRef_colname = NULL,
  mean_decimal = 2,
  sd_decimal = 2,
  n_decimal = 0,
  pct_decimal = 1,
  use_absolute_values_for_diff = FALSE,
  add_n_of_patients_row = TRUE,
  verbose = TRUE
)
```

Arguments

`in_df`	Data frame containing the analytic cohort.
`out_xlsxpath`	Character path for Excel output, or `NULL` to skip file output.
`exposure_var`	Character name of the exposure column.
`exp_value`, `ref_value`	Values identifying the exposed and reference groups. Non-missing values outside these two groups are rejected; missing exposure values are reported and excluded from arm-specific calculations.
`use_weights`	Logical; use `weight_var` for weighted summaries.
`weight_var`	Character name of the weight column. The default is `"psweight"`.
`cont_vars`	Character vector of continuous-variable names.
`cat_vars`	Character vector of categorical-variable names, or a named list whose elements give the levels to display for each variable.
`binary_vars`	Character vector of binary-variable names.
`drop_vars`	Character vector of variables to remove from the requested continuous, binary, and categorical sets.
`drop_varpattern`	Character vector of regular-expression patterns; matching requested variables are removed.
`Var_colname`, `Vartype_colname`, `Total_colname`, `Exp_colname`, `Ref_colname`	Output names for the variable, type, total, exposed, and reference columns.
`CrudeDiff_colname`, `StdDiff_colname`	Output names for the crude- and standardized-difference columns.
`MissingTotal_colname`, `MissingExp_colname`, `MissingRef_colname`	Output names for missing-count and missing-percent columns.

`ExistingTotal_colname`, `ExistingExp_colname`, `ExistingRef_colname`
: Output names for non-missing-count columns. Set a name to `NULL` to omit that column; exposed and reference columns are omitted by default.

`mean_decimal`, `sd_decimal`
: Numbers of decimal places for means and SDs.

`n_decimal`
: Number of decimal places for counts or weighted totals.

`pct_decimal`
: Number of decimal places for percentages.

`use_absolute_values_for_diff`
: Logical; report absolute rather than signed crude differences and SMDs.

`add_n_of_patients_row`
: Logical; prepend an `n_of_patients` row.

`verbose`
: Logical; print progress and validation messages.

Value

Invisibly returns the formatted Table 1 data frame.

Weighted analysis

Setting `use_weights = TRUE` is an explicit contract. `weight_var` must name an existing numeric column whose values are finite, non-missing, and non-negative. Numeric-looking character values are rejected rather than coerced. Zero-weight rows are reported and removed before the survey design and every table statistic are created; an error is raised if no positive-weight rows or no positive-weight rows in one exposure arm remain. A missing weight column is an error, not a silent unweighted fallback.

Weighted continuous summaries use survey-weighted means and variances. Weighted categorical and binary summaries use weighted cell totals and proportions. The `n_of_patients` row reports sums of weights when weighting is enabled.

Meaning of missing and existing counts

In weighted mode, the `Missing_*` columns and variable-row `Existing_*` columns remain unweighted row counts (after zero-weight rows have been removed). They describe data availability, not weighted population size. The exception is the `n_of_patients` row: its Total/Exp/Ref values and its `Existing_*` values are sums of weights. Consequently, an `Existing_Total_N_denom` column mixes an effective weighted total in the `n_of_patients` row with unweighted non-missing row counts in variable rows.

Side effects

With `out_xlsxpath`, creates its parent directory if needed and writes an Excel workbook.

Examples

```
df <- read.csv(system.file("extdata", "sample_data.csv",
                           package = "rwetools"))
tbl <- build_table1(
  in_df = df, exposure_var = "exposure",
```

```
  cont_vars = c("cont1", "cont2", "cont3"),
  binary_vars = c("binary1", "binary2"),
  cat_vars = c("cat1", "cat2"), verbose = FALSE
)
head(tbl)


# Weighted Table 1 with Excel output
if (requireNamespace("openxlsx", quietly = TRUE)) {
  df_ps <- estimate_ps(
    in_df = df, exposure_var = "exposure",
    class_vars = c("cat1", "cat2", "cat3", "cat4"),
    cont_vars = c("cont1", "cont2", "cont3"), verbose = FALSE
  )
  df_wt <- create_matching_weights(
    in_df = df_ps, exposure_var = "exposure", ps_var = "ps",
    weight_var = "mw_wt", verbose = FALSE
  )
  tbl_wt <- build_table1(
    in_df = df_wt, out_xlsxpath = tempfile(fileext = ".xlsx"),
    exposure_var = "exposure", use_weights = TRUE,
    weight_var = "mw_wt", cont_vars = c("cont1", "cont2", "cont3"),
    binary_vars = c("binary1", "binary2"), cat_vars = c("cat1", "cat2"),
    verbose = FALSE
  )
}
```

`create_iptw` *Inverse-probability-of-treatment weights (IPTW, ATE)*

Description

Adds inverse-probability-of-treatment weights targeting the average treatment effect (ATE) to a data set that already contains propensity scores, and optionally writes a diagnostic Excel report, distribution plots, and a weighted/unweighted Table 1. Set `stabilize = TRUE` for stabilized IPTW.

Usage

```
create_iptw(
  in_df = NULL,
  in_csvpath = NULL,
  out_csvpath = NULL,
  out_xlsxpath_report = NULL,
  out_dir_plots = NULL,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
```

```
  ps_var = "ps",
  weight_var = "psweight",
  stabilize = FALSE,
  trim_method = c("none", "crump", "sturmer"),
  trim_crump_alpha = 0.1,
  trim_sturmer_p = 0.05,
  truncate_method = c("none", "percentile", "cap"),
  truncate_percentile = c(0.01, 0.99),
  truncate_cap = NULL,
  make_unwt_wt_table1 = FALSE,
  table1_cont_vars = NULL,
  table1_binary_vars = NULL,
  table1_cat_vars = NULL,
  std_diff_threshold = 0.1,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

`in_df`	Data frame with PS already calculated (optional if `in_csvpath` given).
`in_csvpath`	Character. Path to input CSV with PS already calculated (optional if `in_df` given).
`out_csvpath`	Character. Path for output CSV (optional).
`out_xlsxpath_report`	Character. Path for the Excel diagnostic report (optional; requires **openxlsx**).
`out_dir_plots`	Character. Directory for plot files (optional; requires **ggplot2**).
`exposure_var`	Character. Binary exposure/treatment column (default "exp").
`exp_value`	Value of the exposed/treated group (default 1).
`ref_value`	Value of the reference/control group (default 0).
`ps_var`	Character. PS column name (default "ps").
`weight_var`	Character. Name for the weight column (default "psweight").
`stabilize`	Logical. If TRUE, compute stabilized IPTW (default FALSE).
`trim_method`	Character. PS trimming: "none" (default), "crump" (symmetric, keep PS in `[alpha, 1 - alpha]`) or "sturmer" (asymmetric, exposure-group percentile tails).
`trim_crump_alpha`	Numeric in (0, 0.5). Symmetric Crump bound (default 0.1).
`trim_sturmer_p`	Numeric in (0, 0.5). Sturmer tail percentile (default 0.05).
`truncate_method`	Character. Weight truncation (IPTW only): "none" (default), "percentile" (winsorize to `truncate_percentile`) or "cap" (absolute upper cap `truncate_cap`).
`truncate_percentile`	Length-2 numeric `c(lower, upper)` used when `truncate_method = "percentile"` (default `c(0.01, 0.99)`; upper-only truncation is `c(0, 0.99)`).

`truncate_cap` Single positive number used when `truncate_method = "cap"`.

`make_unwt_wt_table1`
: Logical. Build unweighted and weighted Table 1 (default FALSE; only used when `out_xlsxpath_report` is given).

`table1_cont_vars`
: Character vector. Continuous vars for Table 1 (auto-detected if NULL).

`table1_binary_vars`
: Character vector. Binary vars for Table 1 (auto-detected if NULL).

`table1_cat_vars`
: Character vector. Categorical vars for Table 1 (auto-detected if NULL).

`std_diff_threshold`
: Numeric. Balance threshold on the raw (0-1) standardized-difference scale (default 0.1).

`readme_text` Character. Optional message for a README sheet in the Excel report.

`verbose` Logical. Print progress messages (default TRUE).

Details

This is the ATE member of the rwetools weighting family, which replaces the removed `create_ps_weights()`. Use `create_matching_weights` for matching weights (ATM) and `create_overlap_weights` for overlap weights (ATO).

Value

Invisibly, the input data with the weight column added (rows are removed if trimming is applied). Outputs are written when paths are given.

Estimand

The estimand is fixed at the ATE. IPTW for the ATT (SMR weights: treated = 1, control = PS / (1 - PS)) is a valid method but is not supported in this version.

Trimming and truncation

Optional propensity-score trimming (`trim_method`) and weight truncation (`truncate_method`) are off by default. Both change the analytic population, so trimmed/truncated estimates refer to the trimmed (analytic) population and its estimand, not the original cohort; report them accordingly (typically as sensitivity analyses). `trim_method = "sturmer"` assumes `exp_value` denotes the treated / new-treatment group.

Side Effects

- Writes a CSV file when `out_csvpath` is provided.
- Creates directories, writes an Excel diagnostic report, and saves PNG plot files when the corresponding path arguments are supplied.

Examples

```
csv_path <- system.file("extdata", "sample_data.csv", package = "rwetools")
df_ps <- estimate_ps(
  in_df        = read.csv(csv_path),
  exposure_var = "exposure",
  class_vars   = c("cat1", "cat2", "cat3", "cat4"),
  cont_vars    = c("cont1", "cont2", "cont3"),
  verbose      = FALSE
)

# IPTW (ATE)
df_iptw <- create_iptw(
  in_df        = df_ps,
  exposure_var = "exposure",
  ps_var       = "ps",
  weight_var   = "iptw_wt",
  verbose      = FALSE
)
summary(df_iptw$iptw_wt)

# Stabilized IPTW with upper-only weight truncation at the 99th percentile
df_siptw <- create_iptw(
  in_df               = df_ps,
  exposure_var        = "exposure",
  ps_var              = "ps",
  weight_var          = "siptw_wt",
  stabilize           = TRUE,
  truncate_method     = "percentile",
  truncate_percentile = c(0, 0.99),
  verbose             = FALSE
)
summary(df_siptw$siptw_wt)
```

`create_matching_weights` *Propensity-score matching weights (ATM)*

Description

Adds matching weights (Li & Greene 2013) to a data set that already contains propensity scores, and optionally writes the same diagnostic report, plots, and Table 1 as `create_iptw`. Matching weights target the ATM (the estimand of the matched population) and are bounded in `[0, 1]`, so weight truncation does not apply.

Usage

```
create_matching_weights(
  in_df = NULL,
  in_csvpath = NULL,
```

```
  out_csvpath = NULL,
  out_xlsxpath_report = NULL,
  out_dir_plots = NULL,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
  ps_var = "ps",
  weight_var = "psweight",
  trim_method = c("none", "crump", "sturmer"),
  trim_crump_alpha = 0.1,
  trim_sturmer_p = 0.05,
  make_unwt_wt_table1 = FALSE,
  table1_cont_vars = NULL,
  table1_binary_vars = NULL,
  table1_cat_vars = NULL,
  std_diff_threshold = 0.1,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

`in_df`	Data frame with PS already calculated (optional if `in_csvpath` given).
`in_csvpath`	Character. Path to input CSV with PS already calculated (optional if `in_df` given).
`out_csvpath`	Character. Path for output CSV (optional).
`out_xlsxpath_report`	Character. Path for the Excel diagnostic report (optional; requires **openxlsx**).
`out_dir_plots`	Character. Directory for plot files (optional; requires **ggplot2**).
`exposure_var`	Character. Binary exposure/treatment column (default "exp").
`exp_value`	Value of the exposed/treated group (default 1).
`ref_value`	Value of the reference/control group (default 0).
`ps_var`	Character. PS column name (default "ps").
`weight_var`	Character. Name for the weight column (default "psweight").
`trim_method`	Character. PS trimming: "none" (default), "crump" (symmetric, keep PS in `[alpha, 1 - alpha]`) or "sturmer" (asymmetric, exposure-group percentile tails).
`trim_crump_alpha`	Numeric in (0, 0.5). Symmetric Crump bound (default 0.1).
`trim_sturmer_p`	Numeric in (0, 0.5). Sturmer tail percentile (default 0.05).
`make_unwt_wt_table1`	Logical. Build unweighted and weighted Table 1 (default FALSE; only used when `out_xlsxpath_report` is given).
`table1_cont_vars`	Character vector. Continuous vars for Table 1 (auto-detected if NULL).

`table1_binary_vars`
: Character vector. Binary vars for Table 1 (auto-detected if NULL).

`table1_cat_vars`
: Character vector. Categorical vars for Table 1 (auto-detected if NULL).

`std_diff_threshold`
: Numeric. Balance threshold on the raw (0-1) standardized-difference scale (default 0.1).

`readme_text`
: Character. Optional message for a README sheet in the Excel report.

`verbose`
: Logical. Print progress messages (default TRUE).

Value

Invisibly, the input data with the weight column added (rows are removed if trimming is applied).

Side Effects

- Writes a CSV file when `out_csvpath` is provided.
- Creates directories, writes an Excel diagnostic report, and saves PNG plot files when the corresponding path arguments are supplied.

Examples

```
csv_path <- system.file("extdata", "sample_data.csv", package = "rwetools")
df_ps <- estimate_ps(
  in_df        = read.csv(csv_path),
  exposure_var = "exposure",
  class_vars   = c("cat1", "cat2", "cat3", "cat4"),
  cont_vars    = c("cont1", "cont2", "cont3"),
  verbose      = FALSE
)
df_mw <- create_matching_weights(
  in_df        = df_ps,
  exposure_var = "exposure",
  ps_var       = "ps",
  weight_var   = "mw_wt",
  verbose      = FALSE
)
summary(df_mw$mw_wt)
```

`create_overlap_weights` *Propensity-score overlap weights (ATO)*

Description

Adds overlap weights (Li, Morgan & Zaslavsky 2018) to a data set that already contains propensity scores, and optionally writes the same diagnostic report, plots, and Table 1 as `create_iptw`. Overlap weights target the ATO (average treatment effect in the overlap population) and are bounded in `[0, 1]`, so weight truncation does not apply.

Usage

```
create_overlap_weights(
  in_df = NULL,
  in_csvpath = NULL,
  out_csvpath = NULL,
  out_xlsxpath_report = NULL,
  out_dir_plots = NULL,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
  ps_var = "ps",
  weight_var = "psweight",
  trim_method = c("none", "crump", "sturmer"),
  trim_crump_alpha = 0.1,
  trim_sturmer_p = 0.05,
  make_unwt_wt_table1 = FALSE,
  table1_cont_vars = NULL,
  table1_binary_vars = NULL,
  table1_cat_vars = NULL,
  std_diff_threshold = 0.1,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

`in_df`	Data frame with PS already calculated (optional if `in_csvpath` given).
`in_csvpath`	Character. Path to input CSV with PS already calculated (optional if `in_df` given).
`out_csvpath`	Character. Path for output CSV (optional).
`out_xlsxpath_report`	Character. Path for the Excel diagnostic report (optional; requires **openxlsx**).
`out_dir_plots`	Character. Directory for plot files (optional; requires **ggplot2**).
`exposure_var`	Character. Binary exposure/treatment column (default "exp").
`exp_value`	Value of the exposed/treated group (default 1).
`ref_value`	Value of the reference/control group (default 0).
`ps_var`	Character. PS column name (default "ps").
`weight_var`	Character. Name for the weight column (default "psweight").
`trim_method`	Character. PS trimming: "none" (default), "crump" (symmetric, keep PS in `[alpha, 1 - alpha]`) or "sturmer" (asymmetric, exposure-group percentile tails).
`trim_crump_alpha`	Numeric in (0, 0.5). Symmetric Crump bound (default 0.1).
`trim_sturmer_p`	Numeric in (0, 0.5). Sturmer tail percentile (default 0.05).

`make_unwt_wt_table1`
: Logical. Build unweighted and weighted Table 1 (default FALSE; only used when `out_xlsxpath_report` is given).

`table1_cont_vars`
: Character vector. Continuous vars for Table 1 (auto-detected if NULL).

`table1_binary_vars`
: Character vector. Binary vars for Table 1 (auto-detected if NULL).

`table1_cat_vars`
: Character vector. Categorical vars for Table 1 (auto-detected if NULL).

`std_diff_threshold`
: Numeric. Balance threshold on the raw (0-1) standardized-difference scale (default 0.1).

`readme_text`
: Character. Optional message for a README sheet in the Excel report.

`verbose`
: Logical. Print progress messages (default TRUE).

Value

Invisibly, the input data with the weight column added (rows are removed if trimming is applied).

Side Effects

- Writes a CSV file when `out_csvpath` is provided.
- Creates directories, writes an Excel diagnostic report, and saves PNG plot files when the corresponding path arguments are supplied.

Examples

```
csv_path <- system.file("extdata", "sample_data.csv", package = "rwetools")
df_ps <- estimate_ps(
  in_df        = read.csv(csv_path),
  exposure_var = "exposure",
  class_vars   = c("cat1", "cat2", "cat3", "cat4"),
  cont_vars    = c("cont1", "cont2", "cont3"),
  verbose      = FALSE
)
df_ow <- create_overlap_weights(
  in_df        = df_ps,
  exposure_var = "exposure",
  ps_var       = "ps",
  weight_var   = "ow_wt",
  verbose      = FALSE
)
summary(df_ow$ow_wt)
```

`create_ps_fs_weights` *Create Propensity-Score Fine-Stratification Weights*

Description

Divides the propensity-score distribution into fine strata and assigns ATT or ATE weights from the exposed/reference composition of each retained stratum. It can first restrict the cohort to common propensity-score support and can produce balance tables, plots, and an Excel diagnostic report.

Usage

```
create_ps_fs_weights(
  in_df,
  out_csvpath = NULL,
  out_xlsxpath_report = NULL,
  out_dir_plots = NULL,
  trim_nonoverlap_region = TRUE,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
  ps_var = NULL,
  weight_var = "ps_fs_wt",
  number_of_strata = 50,
  stratification_method = c("exposure", "cohort"),
  estimand = c("ATT", "ATE"),
  make_unwt_wt_table1 = FALSE,
  table1_cont_vars = NULL,
  table1_binary_vars = NULL,
  table1_cat_vars = NULL,
  std_diff_threshold = 0.1,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

`in_df`	Data frame containing exposure and an estimated propensity score.
`out_csvpath`	Character path for the weighted cohort CSV, or `NULL`.
`out_xlsxpath_report`	Character path for an Excel diagnostic report, or `NULL`.
`out_dir_plots`	Character directory for diagnostic PNG files, or `NULL`.
`trim_nonoverlap_region`	Logical; restrict to the intersection of the two arm-specific PS ranges before creating strata.
`exposure_var`	Character name of the exposure column. The default is `"exp"`.

`exp_value`, `ref_value`
: Values identifying the exposed and reference arms.

`ps_var`
: Character name of the propensity-score column. Required.

`weight_var`
: Character name for the created weight column. The default is `"ps_fs_wt"`.

`number_of_strata`
: Positive number of fine strata. The default is 50.

`stratification_method`
: `"exposure"` or `"cohort"`; see the description.

`estimand`
: `"ATT"` or `"ATE"`.

`make_unwt_wt_table1`
: Logical; include crude and weighted balance tables in the diagnostic report.

`table1_cont_vars`, `table1_binary_vars`
: Character vectors of continuous and binary variables for Table 1. When all Table 1 variable arguments are `NULL`, variable types are detected automatically.

`table1_cat_vars`
: Character vector, or named level list, for categorical variables in Table 1.

`std_diff_threshold`
: Absolute SMD threshold used in diagnostics. The default is 0.1 on the raw SMD scale.

`readme_text`
: Optional text for the report's README worksheet.

`verbose`
: Logical; print progress and diagnostic messages.

Details

`stratification_method = "exposure"` forms cut points from the exposed arm; `"cohort"` forms them from the full cohort. Strata without both arms are excluded before weights are assigned. The output keeps the caller's original exposure labels and row alignment, and adds `strata` plus the requested `weight_var`.

Value

Invisibly returns the retained, weighted data frame with `strata` and `weight_var` columns.

Exposure coding

`exp_value` and `ref_value` are always recoded internally to 1 and 0 for trimming, strata construction, weights, and diagnostics. They must be distinct scalar, non-missing values and both must occur. Rows matching neither value are reported and removed. The returned exposure column is restored to its original labels. For a logical exposure, use `TRUE` and `FALSE` rather than numeric 1 and 0.

Using the result in a hazard model

The `strata` column records the PS fine stratum used to construct the weights. Do not automatically pass this column as `strata_var` to `estimate_hr()`. Fine-stratification weighting and `survival::strata()` are distinct adjustments; combining them changes the fitted hazard model and should be an explicit design decision.

Side effects

Writes a CSV, Excel workbook, or diagnostic PNG files when the corresponding output arguments are supplied, creating parent directories as needed.

Examples

```
df <- read.csv(system.file("extdata", "sample_data.csv",
                           package = "rwetools"))
df_ps <- estimate_ps(
  in_df = df, exposure_var = "exposure",
  class_vars = c("cat1", "cat2", "cat3", "cat4"),
  cont_vars = c("cont1", "cont2", "cont3"), verbose = FALSE
)
result <- create_ps_fs_weights(
  in_df = df_ps, exposure_var = "exposure", ps_var = "ps",
  weight_var = "fs_wt", number_of_strata = 10,
  stratification_method = "exposure", estimand = "ATT",
  verbose = FALSE
)
summary(result$fs_wt)
```

---

create_ps_matched_cohort

Create a Propensity-Score-Matched Cohort

Description

Uses `MatchIt::matchit()` with a precomputed propensity score to perform nearest-neighbour or subclass matching. The returned cohort retains the caller's original exposure labels and includes `match_id` and `.match_weights` for downstream balance and effect analysis. Optional outputs provide pre/post-match diagnostics, tables, plots, and CSV files.

Usage

```
create_ps_matched_cohort(
  in_df = NULL,
  in_csvpath = NULL,
  out_csvpath_matcheddata = NULL,
  out_csvpath_crudedata_w_matchindicator = NULL,
  out_xlsxpath_report = NULL,
  out_dir_plots = NULL,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
  ps_var = "ps",
  method = c("nearest", "subclass"),
  ratio = 1,
  min_controls = NULL,
```

```
  max_controls = NULL,
  m_order = NULL,
  subclass_n = NULL,
  caliper = 0.2,
  caliper_scale = c("logit_ps_sd", "raw", "raw_ps_sd"),
  replace = FALSE,
  trim_method = c("none", "crump", "sturmer"),
  trim_crump_alpha = 0.1,
  trim_sturmer_p = 0.05,
  make_crude_matched_table1 = FALSE,
  table1_cont_vars = NULL,
  table1_binary_vars = NULL,
  table1_cat_vars = NULL,
  std_diff_threshold = 0.1,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

`in_df`	Data frame containing a precomputed PS, or `NULL` when `in_csvpath` is used.
`in_csvpath`	Character path to an input CSV, or `NULL`. If both inputs are supplied, `in_df` is used with a warning.
`out_csvpath_matcheddata`	Character path for the matched-cohort CSV, or `NULL`.
`out_csvpath_crudedata_w_matchindicator`	Character path for the input cohort augmented with matching indicators, or `NULL`.
`out_xlsxpath_report`	Character path for an Excel diagnostic report, or `NULL`.
`out_dir_plots`	Character directory for diagnostic PNG files, or `NULL`.
`exposure_var`	Character name of the exposure column. The default is `"exp"`.
`exp_value, ref_value`	Values identifying the exposed and reference arms.
`ps_var`	Character name of the propensity-score column.
`method`	`"nearest"` or `"subclass"`. Optimal and full matching are not supported.
`ratio`	Nearest-neighbour control ratio, 1:k. The default is 1.
`min_controls, max_controls`	Optional variable-ratio bounds passed as `min.controls` and `max.controls` for nearest matching.
`m_order`	Optional nearest-matching order passed as `m.order`.
`subclass_n`	Optional number of subclasses. `NULL` uses MatchIt's default.
`caliper`	Positive caliper width interpreted on `caliper_scale`. The default is 0.2.
`caliper_scale`	`"logit_ps_sd"`, `"raw_ps_sd"`, or `"raw"`; see **Caliper scales**.
`replace`	Logical; perform nearest matching with replacement.

`trim_method` PS trimming before matching: `"none"`, `"crump"`, or `"sturmer"`. Trimming changes the analytic population.

`trim_crump_alpha`
Crump symmetric bound in (0, 0.5).

`trim_sturmer_p` Sturmer arm-specific tail proportion in (0, 0.5).

`make_crude_matched_table1`
Logical; include crude and matched Table 1 sheets in the diagnostic report.

`table1_cont_vars`, `table1_binary_vars`
Character vectors of continuous and binary variables for Table 1. When all Table 1 variable arguments are `NULL`, types are detected automatically.

`table1_cat_vars`
Character vector, or named level list, for categorical variables in Table 1.

`std_diff_threshold`
Absolute SMD threshold for diagnostics. The default is 0.1 on the raw SMD scale.

`readme_text` Optional text for the report's README worksheet.

`verbose` Logical; print progress and diagnostic messages.

Value

Invisibly returns the matched-cohort data frame. It includes `matched`, `match_id`, and `.match_weights`; internal recoding columns are removed.

Exposure coding

`exp_value` and `ref_value` are always recoded internally to 1 and 0 for MatchIt. They must be distinct scalar, non-missing values and both must occur. Rows matching neither value are reported and removed. The returned exposure column is restored to its original labels and row alignment. For a logical exposure, use `TRUE` and `FALSE`, not numeric 1 and 0.

Caliper scales

- `"logit_ps_sd"` (default): match on logit(PS) with a caliper equal to `caliper` times SD(logit(PS)); PS must lie strictly inside (0, 1).
- `"raw_ps_sd"`: match on PS with a caliper equal to `caliper` times SD(PS). This reproduces the rwetools 0.1.x scale.
- `"raw"`: match on PS with an absolute raw-PS caliper.

Calipers apply to nearest matching, not subclass matching.

Choosing the downstream effect block

A no-replacement nearest 1:1 cohort with unit `.match_weights` can be passed to `in_df_match` in `estimate_hr()`, `estimate_ir()`, or `estimate_risk()`; pass `match_id` as `if_match_match_id` for set-aware inference where the effect function supports it.

Other designs are weight-defined. Subclass matching returns subclass weights. Variable-ratio matching, and fixed `ratio` > 1 when some exposed units find fewer than the requested controls, can

also return non-uniform control weights. Pass these cohorts as `in_df_weight` with `if_weight_weight_var = ".match_weights"` to honour the weights. This route does not preserve matched-set clustering; see the effect-function documentation for the resulting inference limitations.

The same weights are required for balance assessment. The built-in matched Table 1 automatically applies `.match_weights` whenever the returned values are non-unit, including subclass and variable/incomplete nearest 1:k output.

With `replace = TRUE`, a reused control can belong to several matched sets; its `match_id` contains the joined set ids. Such output is rejected by the effect matched block. The weighted route honours reuse weights but is not an Abadie-Imbens replacement variance, and this function does not return the one-row-per-unit-per-pair representation needed for pair-aware replacement inference.

Side effects

Writes matched/crude CSV files, an Excel report, or PNG figures when the corresponding output arguments are supplied, creating parent directories as needed.

Examples

```
if (requireNamespace("MatchIt", quietly = TRUE)) {
  df <- read.csv(system.file("extdata", "sample_data.csv",
                             package = "rwetools"))
  df_ps <- estimate_ps(
    in_df = df, exposure_var = "exposure",
    class_vars = c("cat1", "cat2", "cat3", "cat4"),
    cont_vars = c("cont1", "cont2", "cont3"), verbose = FALSE
  )
  matched <- create_ps_matched_cohort(
    in_df = df_ps, exposure_var = "exposure", ps_var = "ps",
    ratio = 1, caliper = 0.2, verbose = FALSE
  )
  nrow(matched)
}
```

`estimate_hr` *Estimate a Cox Hazard Ratio or Fine-Gray Subdistribution Hazard Ratio*

Description

Fits either a Cox proportional-hazards model or a Fine-Gray model for a binary exposure. A call can contain a crude cohort plus either a weighted cohort or a matched cohort. The weighted and matched blocks are mutually exclusive, and the crude block is optional.

Usage

```
estimate_hr(
  in_df_crude = NULL,
  in_df_weight = NULL,
  in_df_match = NULL,
  out_xlsxpath = NULL,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
  outcome_var = NULL,
  followuptime_var = NULL,
  confidence_level = 0.95,
  strata_var = NULL,
  hr_model = c("Cox", "Fine-Gray"),
  if_fg_competing_event_var = NULL,
  if_weight_weight_var = NULL,
  if_match_match_id = NULL,
  if_bootstrap_count = NULL,
  if_bootstrap_n_cores = NULL,
  if_bootstrap_seed = NULL,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

`in_df_crude`	Data frame for a crude block, or `NULL`.
`in_df_weight`	Data frame for a weighted block, or `NULL`. Requires `if_weight_weight_var` and cannot be combined with `in_df_match`.
`in_df_match`	Data frame for a matched block, or `NULL`. Cannot be combined with `in_df_weight`.
`out_xlsxpath`	Character path for an Excel workbook, or `NULL`.
`exposure_var`	Character name of the exposure column. The default is `"exp"`.
`exp_value, ref_value`	Values identifying the exposed and reference arms in `exposure_var`.
`outcome_var`	Character name of the event indicator, coded 1 for the event of interest.
`followuptime_var`	Character name of the follow-up time column. The Cox/Fine-Gray coefficient is invariant to a constant change of time unit.
`confidence_level`	Single number strictly between 0 and 1. The default is 0.95.
`strata_var`	Character name of a model-stratification column, or `NULL`. See **Stratification is not standardization**.
`hr_model`	`"Cox"` for a cause-specific HR or `"Fine-Gray"` for a subdistribution HR.
`if_fg_competing_event_var`	Character name of a binary competing-event indicator, coded 1 for a competing event. Required for Fine-Gray and ignored with a message for Cox.

`if_weight_weight_var`
: Character name of the analysis-weight column in `in_df_weight`. Required for a weighted block and ignored with a message otherwise.

`if_match_match_id`
: Character name of the set id in `in_df_match`. When supplied, it defines the robust-SE cluster and pair-level bootstrap unit. Ignored with a message without a matched block.

`if_bootstrap_count`
: Positive number of percentile-bootstrap replicates, or `NULL` to omit bootstrap CIs. Weights are frozen at their supplied values, so PS-estimation uncertainty is not propagated.

`if_bootstrap_n_cores`
: Number of bootstrap workers, or `NULL` to use all detected cores minus one. Ignored without `if_bootstrap_count`.

`if_bootstrap_seed`
: Integer RNG seed, or `NULL`. It makes this function reproducible, but does not coordinate draws with `estimate_ir()`. Ignored without `if_bootstrap_count`.

`readme_text`
: Optional text for a README worksheet when Excel output is requested.

`verbose`
: Logical; print progress, exclusions, and method messages.

Value

Invisibly returns a list. Cox analysis supplies `hazard_ratios` and per-block `models`; Fine-Gray supplies `subdist_hazard`. Bootstrap draws are in `bootstrap` when requested, and `strata_var` is returned when used.

Stratification is not standardization

`strata_var` adds `survival::strata()` to the hazard model. It allows a separate baseline hazard in each level while estimating one conditional common exposure (s)HR. This is not a marginal standardized rate or risk. Every retained level must contain both exposure arms; the function stops otherwise because a single-arm level contributes no within-level comparison to the partial likelihood.

Do not automatically pass the `strata` column produced by `create_ps_fs_weights()` as `strata_var`. Fine-stratification weights and a stratified hazard model are different adjustments; using both is a separate analysis choice that changes the fitted hazard model.

Exposure coding and analysis rows

`exp_value` and `ref_value` are always recoded internally to 1 and 0, including when the source column is already coded 0/1. They must be distinct scalar, non-missing values. Rows matching neither value are reported and removed; rows incomplete on variables used by the requested model are also removed. Both arms must remain. For a logical exposure column, use `exp_value = TRUE` and `ref_value = FALSE`, not numeric 1 and 0.

Analytical confidence intervals

Methods are fixed by block; there are no CI-method arguments.

- Cox: crude uses the model SE, weighted uses a robust sandwich SE, and matched uses a robust SE clustered on `if_match_match_id` when the id is supplied.
- Fine-Gray: the expanded-data Cox fit uses robust inference clustered on subject for crude and weighted analyses, or on match id for a matched analysis. Weighted expansion rows carry the product of IPCW and the supplied analysis weight.

Matched designs and matching weights

The matched block is an unweighted matched analysis. If a `.match_weights` column is present, every retained value must equal 1. With `if_match_match_id`, every complete-case set must contain exactly one exposed and one reference row; matching with replacement is rejected because a reused unit belongs to multiple sets. Without a match id, inference is not clustered by matched set and the bootstrap resamples rows.

For subclass, variable-ratio, fixed ratios greater than 1, or with-replacement matching, pass the cohort as `in_df_weight` and set `if_weight_weight_var = ".match_weights"` to honour the matching weights. That route provides a weight-based robust SE but does not use matched-set clustering. It therefore cannot express weights and set clustering together, and it is not the Abadie-Imbens variance for matching with replacement. The package also does not produce the one-row-per-unit-per-pair representation required for design-aware replacement inference.

Migration from rwetools 0.4.0

`estimate_hr_ir()` was split into `estimate_hr()` and `estimate_ir()`. The old `stratification_var` argument is now `strata_var` here only. `time_unit` and `ir_per_pyears` belong to `estimate_ir()` and are not arguments to this function. With `hr_model = "Fine-Gray"`, missing values in the competing event variable affect this hazard analysis but no longer remove rows from a separate incidence-rate analysis.

Side effects

With `out_xlsxpath`, creates its parent directory if needed and writes an Excel workbook. With `verbose = TRUE`, prints progress and method messages.

See Also

`estimate_ir()`, `estimate_risk()`, `create_ps_matched_cohort()`

Examples

```
df <- read.csv(system.file("extdata", "sample_data.csv",
                           package = "rwetools"))
res <- estimate_hr(
  in_df_crude      = df,
  exposure_var     = "exposure",
  outcome_var      = "outcome",
  followuptime_var = "follow_up_days",
```

```
  verbose          = FALSE
)
res$hazard_ratios


# Crude and weighted Fine-Gray blocks
df_ps <- estimate_ps(
  in_df = df, exposure_var = "exposure",
  class_vars = c("cat1", "cat2", "cat3", "cat4"),
  cont_vars = c("cont1", "cont2", "cont3"), verbose = FALSE
)
df_wt <- create_iptw(
  in_df = df_ps, exposure_var = "exposure", ps_var = "ps",
  weight_var = "iptw", verbose = FALSE
)
res_fg <- estimate_hr(
  in_df_crude = df, in_df_weight = df_wt,
  if_weight_weight_var = "iptw", exposure_var = "exposure",
  outcome_var = "outcome", followuptime_var = "follow_up_days",
  hr_model = "Fine-Gray",
  if_fg_competing_event_var = "competing_event", verbose = FALSE
)
res_fg$subdist_hazard
```

`estimate_ir` *Estimate Marginal Incidence Rates, Rate Differences, and Rate Ratios*

Description

Calculates incidence rates (IR), exposed-minus-reference rate differences (IRD), and exposed-versus-reference rate ratios (IRR). A call can contain a crude cohort plus either a weighted cohort or a matched cohort. The weighted and matched blocks are mutually exclusive, and the crude block is optional.

Usage

```
estimate_ir(
  in_df_crude = NULL,
  in_df_weight = NULL,
  in_df_match = NULL,
  out_xlsxpath = NULL,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
  outcome_var = NULL,
  followuptime_var = NULL,
  time_unit = c("days", "months", "years"),
  ir_per_pyears = 1000,
```

```
  confidence_level = 0.95,
  if_weight_weight_var = NULL,
  if_match_match_id = NULL,
  if_bootstrap_count = NULL,
  if_bootstrap_n_cores = NULL,
  if_bootstrap_seed = NULL,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

`in_df_crude`	Data frame for a crude block, or `NULL`.
`in_df_weight`	Data frame for a weighted block, or `NULL`. Requires `if_weight_weight_var` and cannot be combined with `in_df_match`.
`in_df_match`	Data frame for a matched block, or `NULL`. Cannot be combined with `in_df_weight`.
`out_xlsxpath`	Character path for an Excel workbook, or `NULL`.
`exposure_var`	Character name of the exposure column. The default is `"exp"`.
`exp_value, ref_value`	Values identifying the exposed and reference arms in `exposure_var`.
`outcome_var`	Character name of the event indicator, coded 1 for the event of interest.
`followuptime_var`	Character name of the follow-up time column.
`time_unit`	Unit of `followuptime_var`: `"days"`, `"months"`, or `"years"`.
`ir_per_pyears`	Reporting multiplier for IR and IRD. Must be one of 1, 100, 1000, 10000, or 100000.
`confidence_level`	Single number strictly between 0 and 1. The default is 0.95.
`if_weight_weight_var`	Character name of the analysis-weight column in `in_df_weight`. Required for a weighted block and ignored with a message otherwise.
`if_match_match_id`	Character name of the set id in `in_df_match`. When supplied, it defines the analytical cluster and pair-level bootstrap unit. Without it, matched rows are treated as independent analytically and resampled by row. Ignored with a message without a matched block.
`if_bootstrap_count`	Positive number of percentile-bootstrap replicates, or `NULL` to omit bootstrap CIs. Weights are frozen at their supplied values, so PS-estimation uncertainty is not propagated.
`if_bootstrap_n_cores`	Number of bootstrap workers, or `NULL` to use all detected cores minus one. Ignored without `if_bootstrap_count`.
`if_bootstrap_seed`	Integer RNG seed, or `NULL`. It makes this function reproducible, but does not coordinate draws with `estimate_hr()`. Ignored without `if_bootstrap_count`.

readme_text	Optional text for a README worksheet when Excel output is requested.
verbose	Logical; print progress, exclusions, and method messages.

Details

Direct standardization was removed in rwetools 0.5.0. This function has no stratification argument: every reported rate measure is marginal over the rows in its analysis block. To address a baseline stratifier, include it in the propensity-score model; alternatively, run separate analyses within levels and combine them explicitly in caller code under a prespecified target-population rule.

Value

Invisibly returns a list with incidence_rates, incidence_rate_ratios, and ir_per_pyears. Per-block bootstrap draw matrices are returned in bootstrap when requested.

Exposure coding and analysis rows

exp_value and ref_value are always recoded internally to 1 and 0, including when the source column is already coded 0/1. They must be distinct scalar, non-missing values. Rows matching neither value are reported and removed; rows incomplete on variables used by this rate analysis are also removed. Both arms must remain. For a logical exposure column, use exp_value = TRUE and ref_value = FALSE, not numeric 1 and 0.

Follow-up is converted to person-years using 365.25 days per year, 12 months per year, or the supplied years directly. The IR and IRD are multiplied by ir_per_pyears; the IRR is unitless.

Analytical confidence intervals

Methods are fixed by block; there are no CI-method arguments.

- Crude: the arm-specific IR uses a Garwood exact-Poisson interval; IRD uses the independent-Poisson variance with a normal-Wald interval; IRR uses the model SE from a marginal Poisson rate model.
- Weighted: IR, IRD, and IRR are contrasts from one saturated weighted quasi-Poisson cell model with a design-based sandwich covariance matrix.
- Matched with if_match_match_id: the same cell model declares the matched set as the sampling unit, so all three intervals are cluster-robust. The point estimates do not change. The intervals may be narrower or wider than independence intervals, depending on the sign of within-set covariance.
- Matched without if_match_match_id: rows are treated as independent and the crude analytical methods are used.

Matched designs and matching weights

The matched block is an unweighted matched analysis. If a .match_weights column is present, every retained value must equal 1. With if_match_match_id, every complete-case set must contain exactly one exposed and one reference row; matching with replacement is rejected.

For subclass, variable-ratio, fixed ratios greater than 1, or with-replacement matching, pass the cohort as in_df_weight and set if_weight_weight_var = ".match_weights". This honours the

weights and uses weight-based robust inference, but it does not retain matched-set clustering. In particular, variable-ratio matching would require weights and set clustering together, which this block API cannot express, and the weighted route is not the Abadie-Imbens variance for replacement matching. The package does not produce a one-row-per-unit-per-pair replacement dataset.

Migration from rwetools 0.4.0

`estimate_hr_ir()` was split into `estimate_hr()` and `estimate_ir()`. Hazard-ratio arguments moved to `estimate_hr()`. `stratification_var` was removed from the rate API together with direct standardization. Fine-Gray competing-event missingness no longer narrows the incidence-rate rows.

Side effects

With `out_xlsxpath`, creates its parent directory if needed and writes an Excel workbook. With `verbose = TRUE`, prints progress and method messages.

See Also

`estimate_hr()`, `estimate_risk()`, `create_ps_matched_cohort()`

Examples

```
df <- read.csv(system.file("extdata", "sample_data.csv",
                           package = "rwetools"))
res <- estimate_ir(
  in_df_crude      = df,
  exposure_var     = "exposure",
  outcome_var      = "outcome",
  followuptime_var = "follow_up_days",
  time_unit        = "days",
  verbose          = FALSE
)
res$incidence_rates
res$incidence_rate_ratios


# Crude and weighted blocks with frozen-weight bootstrap CIs
df_ps <- estimate_ps(
  in_df = df, exposure_var = "exposure",
  class_vars = c("cat1", "cat2", "cat3", "cat4"),
  cont_vars = c("cont1", "cont2", "cont3"), verbose = FALSE
)
df_wt <- create_iptw(
  in_df = df_ps, exposure_var = "exposure", ps_var = "ps",
  weight_var = "iptw", verbose = FALSE
)
res_boot <- estimate_ir(
  in_df_crude = df, in_df_weight = df_wt,
  if_weight_weight_var = "iptw", exposure_var = "exposure",
  outcome_var = "outcome", followuptime_var = "follow_up_days",
  if_bootstrap_count = 200, if_bootstrap_n_cores = 1,
```

```
  if_bootstrap_seed = 2026, verbose = FALSE
)
res_boot$incidence_rates
```

`estimate_ps` *Calculate propensity scores and add them to the dataset*

Description

This function calculates propensity scores using logistic regression and adds them as a new column to the dataset. It can output both an R data frame and/or CSV file. Additionally, it can save odds ratio table from the PS model to Excel or data frame.

Usage

```
estimate_ps(
  in_df = NULL,
  in_csvpath = NULL,
  out_csvpath = NULL,
  out_xlsxpath_odds_ratio = NULL,
  exposure_var = "exposure",
  exp_value = 1,
  ref_value = 0,
  class_vars = NULL,
  cont_vars = NULL,
  ps_var = "ps",
  interactions = NULL,
  exclude_vars_w_extreme_distribution = FALSE,
  separation_action = c("warn", "error", "ignore"),
  verbose = TRUE
)
```

Arguments

`in_df`	Data frame containing the input data (optional if in_csvpath provided)
`in_csvpath`	Character string. Path to input CSV file (optional if in_df provided)
`out_csvpath`	Character string. Path for output CSV file (optional, if NULL no CSV is saved)
`out_xlsxpath_odds_ratio`	Character string. Path for output Excel file with OR table (optional)
`exposure_var`	Character string. Name of the binary exposure/treatment variable column (default: "exposure")
`exp_value`	Value representing the exposed/treated group (default: 1)
`ref_value`	Value representing the reference/control group (default: 0)
`class_vars`	Character vector. Names of categorical/factor variables to include in PS model

`cont_vars` Character vector. Names of continuous variables to include in PS model

`ps_var` Character string. Name for the calculated PS variable column (default: "ps")

`interactions` Character string. Interaction terms to include (e.g., "var1:var2 + var1:var3")

`exclude_vars_w_extreme_distribution`
Logical. If TRUE, automatically excludes variables with extreme distribution (categorical: only 1 unique level in either group, or any level with count < 5 in either group; continuous: constant/single unique value (SD < 1e-6) in either group, or SMD > 1.5 between groups) instead of throwing error (default: FALSE). The categorical screen unions levels across the two exposure groups, so a level present in one group but absent in the other is counted as n = 0 and flagged by the cnt < 5 rule.

`separation_action`
Character. Action taken when the post-fit joint-design (quasi-)separation diagnostic flags the PS model: `"warn"` (default; emit a warning and continue), `"error"` (stop with an error), or `"ignore"` (proceed silently; base-R glm separation warnings are suppressed too). The diagnostic is a base-R heuristic (glm non-convergence, fitted probabilities pinned within 1e-8 of 0 or 1, or a large coefficient paired with a large standard error) that complements the marginal, per-variable extreme-distribution screen (`exclude_vars_w_extreme_distribution`), which is blind to separation arising jointly across covariates. Choosing `"warn"` preserves the prior return value and control flow.

`verbose` Logical. Print progress messages (default TRUE).

Details

The default exposure column here is `"exposure"`, whereas downstream weighting, matching, and effect functions default to `"exp"`. This difference is retained for compatibility; pass `exposure_var` explicitly throughout a pipeline when the source column uses another name.

Value

A data frame with the PS column added (with the same number of rows as the input). Rows with a missing exposure value or a missing PS-model covariate are excluded from model fitting (`na.action = na.exclude`) and receive `NA` for the propensity score, with a warning; all other rows are unchanged. Also saves to CSV if a path is specified.

Side Effects

- Writes a CSV file when `out_csvpath` is provided.
- Writes an Excel file with odds-ratio table when `out_xlsxpath_odds_ratio` is provided.

Examples

```
csv_path <- system.file("extdata", "sample_data.csv", package = "rwetools")
df <- read.csv(csv_path)

# Basic usage: estimate PS and add it as a new column
result <- estimate_ps(
```

```
  in_df        = df,
  exposure_var = "exposure",
  class_vars   = c("cat1", "cat2", "cat3", "cat4"),
  cont_vars    = c("cont1", "cont2", "cont3"),
  verbose      = FALSE
)
head(result$ps)


# With CSV output and Excel OR table (requires openxlsx)
if (requireNamespace("openxlsx", quietly = TRUE)) {
  out_csv  <- tempfile(fileext = ".csv")
  out_xlsx <- tempfile(fileext = ".xlsx")
  result2 <- estimate_ps(
    in_df                               = df,
    out_csvpath                         = out_csv,
    out_xlsxpath_odds_ratio             = out_xlsx,
    exposure_var                        = "exposure",
    class_vars                          = c("cat1", "cat2"),
    cont_vars                           = c("cont1", "cont2"),
    exclude_vars_w_extreme_distribution = TRUE,
    verbose                             = FALSE
  )
}
```

`estimate_risk` *Estimate Marginal Risks, Risk Ratios, and Risk Differences*

Description

Estimates arm-specific cumulative incidence, an exposed-versus-reference risk ratio (RR), and an exposed-minus-reference risk difference (RD) at one time point. A call can contain a crude cohort plus either a weighted cohort or a matched cohort. The weighted and matched blocks are mutually exclusive, and the crude block is optional.

Usage

```
estimate_risk(
  in_df_crude = NULL,
  in_df_weight = NULL,
  in_df_match = NULL,
  out_xlsxpath = NULL,
  exposure_var = "exp",
  exp_value = 1,
  ref_value = 0,
  outcome_var = NULL,
  followuptime_var = NULL,
  time_unit = c("days", "months", "years"),
```

```
  risk_at_timepoint = 365,
  risk_per_individuals = 1000,
  confidence_level = 0.95,
  risk_estimator = c("KM", "AJ"),
  if_aj_competing_event_var = NULL,
  if_weight_weight_var = NULL,
  if_match_match_id = NULL,
  if_bootstrap_count = NULL,
  if_bootstrap_n_cores = NULL,
  if_bootstrap_seed = NULL,
  readme_text = NULL,
  verbose = TRUE
)
```

Arguments

in_df_crude	Data frame for a crude block, or NULL.
in_df_weight	Data frame for a weighted block, or NULL. Requires if_weight_weight_var and cannot be combined with in_df_match.
in_df_match	Data frame for a matched block, or NULL. Cannot be combined with in_df_weight.
out_xlsxpath	Character path for an Excel workbook, or NULL.
exposure_var	Character name of the exposure column. The default is "exp".
exp_value, ref_value	Values identifying the exposed and reference arms in exposure_var.
outcome_var	Character name of the event indicator, coded 1 for the event of interest.
followuptime_var	Character name of the follow-up time column.
time_unit	Unit shared by followuptime_var and risk_at_timepoint: "days", "months", or "years".
risk_at_timepoint	Single positive time at which risk, RR, and RD are evaluated. The default is 365.
risk_per_individuals	Positive reporting multiplier for arm risks and RD. The default is 1000.
confidence_level	Single number strictly between 0 and 1. The default is 0.95.
risk_estimator	"KM" or "Kaplan-Meier"; or "AJ" or "Aalen-Johansen" for competing risks.
if_aj_competing_event_var	Character name of a binary competing-event indicator, coded 1 for a competing event. Required for AJ and ignored with a message for KM.
if_weight_weight_var	Character name of the analysis-weight column in in_df_weight. Required for a weighted block and ignored with a message otherwise.
if_match_match_id	Character name of the set id in in_df_match. It defines the pair-bootstrap unit but does not alter analytical risk, RR, or RD variance. Without it, bootstrap resampling is by row. Ignored with a message without a matched block.

`if_bootstrap_count`
: Positive number of percentile-bootstrap replicates, or `NULL` to omit bootstrap CIs. Weights are frozen at their supplied values.

`if_bootstrap_n_cores`
: Number of bootstrap workers, or `NULL` to use all detected cores minus one. Ignored without `if_bootstrap_count`.

`if_bootstrap_seed`
: Integer RNG seed, or `NULL`. Ignored without `if_bootstrap_count`.

`readme_text`
: Optional text for the Excel README worksheet.

`verbose`
: Logical; print progress, exclusions, and method messages.

Details

Direct standardization was removed in rwetools 0.5.0. This function has no stratification argument: every reported risk measure is marginal over the rows in its analysis block. To address a baseline stratifier, include it in the propensity-score model; alternatively, run separate analyses within levels and combine them explicitly in caller code under a prespecified target-population rule.

`risk_estimator = "KM"` reports `1 - S(t)` from Kaplan-Meier. `risk_estimator = "AJ"` reports the Aalen-Johansen cumulative incidence function and requires a separate binary competing-event indicator.

Value

Invisibly returns a list with `estimates` (one row per block) and `cumulative_incidence` (one row per arm and block). When requested, per-block bootstrap matrices are returned as `crude_bootstrap`, `weighted_bootstrap`, or `matched_bootstrap` for the blocks present.

Exposure coding and analysis rows

`exp_value` and `ref_value` are always recoded internally to 1 and 0, including when the source column is already coded 0/1. They must be distinct scalar, non-missing values. Rows matching neither value are reported and removed; rows incomplete on variables used by this risk analysis are also removed. Both arms must remain. For a logical exposure column, use `exp_value = TRUE` and `ref_value = FALSE`, not numeric 1 and 0.

For Aalen-Johansen, the competing-event indicator must be binary after complete-case removal, and a row cannot have both indicators equal to 1.

Analytical confidence intervals

Methods are fixed; there are no CI-method arguments. Risk/CIF intervals use a complementary log-log transformation of the estimated risk and its Greenwood (KM) or counting-process (AJ) SE. At a risk of exactly 0 or 1 the transform is undefined, so the bounds are `NA`. RR uses a log-scale delta interval and RD uses a normal-Wald interval.

The analytical variance in a matched block treats rows as independent. It does not model within-pair covariance, even when `if_match_match_id` is supplied, and is typically mildly conservative for 1:1 matching. Request a bootstrap and supply `if_match_match_id` for pair-aware resampling.

Matched designs and matching weights

The matched block is an unweighted matched analysis. If a `.match_weights` column is present, every retained value must equal 1. With `if_match_match_id`, every complete-case set must contain exactly one exposed and one reference row; matching with replacement is rejected.

For subclass, variable-ratio, fixed ratios greater than 1, or with-replacement matching, pass the cohort as `in_df_weight` and set `if_weight_weight_var = ".match_weights"` to honour the point-estimate weights. This route still uses fixed-case-weight Greenwood or Aalen-Johansen variance for risk measures and does not use matched-set clustering. It therefore cannot express weights and set clustering together and is not design-aware inference for variable-ratio, subclass, or replacement matching. The package does not produce a one-row-per-unit-per-pair replacement dataset.

Bootstrap

`if_bootstrap_count` adds percentile intervals. Supplied analysis weights are frozen, so propensity-score estimation uncertainty is not propagated. A matched block with `if_match_match_id` resamples whole matched sets; without the id it resamples rows. The seed controls this function only and does not establish shared draws with `estimate_hr()` or `estimate_ir()`.

Migration from rwetools 0.4.0

`estimate_rr_rd()` is now `estimate_risk()`; `rr_rd_at_timepoint` is now `risk_at_timepoint`. The old `stratification_var` and direct-standardization output, including the `Stratified_By` column and stratum-details sheet, were removed.

Side effects

With `out_xlsxpath`, creates its parent directory if needed and writes an Excel workbook containing README, analysis-summary, and cumulative-incidence sheets. With `verbose = TRUE`, prints progress and method messages.

See Also

`estimate_hr()`, `estimate_ir()`, `create_ps_matched_cohort()`

Examples

```
df <- read.csv(system.file("extdata", "sample_data.csv",
                           package = "rwetools"))
res <- estimate_risk(
  in_df_crude = df, exposure_var = "exposure",
  outcome_var = "outcome", followuptime_var = "follow_up_days",
  time_unit = "days", risk_at_timepoint = 365,
  risk_per_individuals = 1000, verbose = FALSE
)
res$estimates


# Aalen-Johansen competing-risk estimates with frozen-weight bootstrap CIs
res_aj <- estimate_risk(
  in_df_crude = df, exposure_var = "exposure",
```

```
  outcome_var = "outcome", followuptime_var = "follow_up_days",
  risk_at_timepoint = 365, risk_estimator = "AJ",
  if_aj_competing_event_var = "competing_event",
  if_bootstrap_count = 100, if_bootstrap_n_cores = 1,
  if_bootstrap_seed = 2026, verbose = FALSE
)
res_aj$estimates
```

Index