Package 'rwetools'

August 1, 2026

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

Version 0.4.0

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, hazard ratios, risk ratios, and risk differences with support for stratified and direct-standardized analyses. All core functions produce formatted 'Excel' reports with embedded 'README' 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 2026-08-01 16:50:02 UTC

Contents

`build_table1` *Build a Table 1 comparing baseline characteristics between groups*

Description

Generates a descriptive Table 1 with counts, percentages, means, SDs, crude differences, standardized mean differences (SMD), and missingness. Supports inverse probability weighting via a user-supplied weight column and optionally saves the output to an Excel file.

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 string. File path for Excel output, or NULL to skip (default NULL).
exposure_var	Character string. Name of the binary exposure column.
exp_value	Value in exposure_var representing the exposed group (default 1).
ref_value	Value in exposure_var representing the reference group (default 0).
use_weights	Logical. Apply weights? (default FALSE).
weight_var	Character string. Name of the weight column (default "psweight").
cont_vars	Character vector of continuous variable names.
cat_vars	Character vector or named list of categorical variable names. If a named list, each element should be a character vector of factor levels.
binary_vars	Character vector of binary variable names.
drop_vars	Character vector of variable names to exclude.
drop_varpattern	Character vector of regex patterns; matching variables are excluded.
Var_colname, Vartype_colname, Total_colname, Exp_colname, Ref_colname	Column-name overrides for the output table.
CrudeDiff_colname, StdDiff_colname	Column-name overrides for difference columns.
MissingTotal_colname, MissingExp_colname, MissingRef_colname	Column-name overrides for missingness columns.
ExistingTotal_colname, ExistingExp_colname, ExistingRef_colname	Column-name overrides for non-missing-count columns. Set to NULL to omit (default for Exp/Ref).
mean_decimal, sd_decimal	Integer. Decimal places for mean and SD.
n_decimal	Integer. Decimal places for counts (default 0).
pct_decimal	Integer. Decimal places for percentages (default 1).
use_absolute_values_for_diff	Logical. Show absolute differences? (default FALSE).
add_n_of_patients_row	Logical. Prepend an "n_of_patients" row? (default TRUE).
verbose	Logical. Print progress messages (default TRUE).

Value

Invisibly returns the Table 1 data frame.

Side Effects

When `out_xlsxpath` is not `NULL`, creates the output directory (if needed) and writes an Excel workbook.

Examples

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

# Unweighted Table 1
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 (requires openxlsx)
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
  )
  out_xlsx <- tempfile(fileext = ".xlsx")
  tbl_wt <- build_table1(
    in_df        = df_wt,
    out_xlsxpath = out_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 *Calculate PS Fine Stratification Weights, Trim Non-overlapping Regions, and Generate Diagnostics*

Description

Combined function that (1) creates PS fine strata, (2) calculates stratification weights, (3) trims non-overlapping PS regions, and optionally (4) builds a crude vs weighted balance table (Table 1) and (5) generates diagnostic plots.

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 with PS already calculated (the "crude" / untrimmed data)
out_csvpath	Character. Path for output CSV of trimmed+weighted data (optional)
out_xlsxpath_report	Character. Path for Excel diagnostic report (optional)
out_dir_plots	Character. Directory to save diagnostic plots (NULL = skip plots)
trim_nonoverlap_region	Logical. Trim non-overlapping PS regions (default TRUE)
exposure_var	Character. Name of binary exposure variable (default "exp")
exp_value	Value representing exposed group (default 1)
ref_value	Value representing reference group (default 0)
ps_var	Character. Name of PS variable in the data (required)
weight_var	Character. Name for the stratification weight variable (default "ps_fs_wt")
number_of_strata	Integer. Number of strata to create (default 50)
stratification_method	Character. "exposure" or "cohort" (default "exposure")
estimand	Character. "ATT" or "ATE" (default "ATT")
make_unwt_wt_table1	Logical. Build crude vs weighted balance table (default FALSE)
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. Threshold for acceptable standardised difference (default 0.1)
readme_text	Character. Optional message for README sheet in Excel report
verbose	Logical. Print progress messages (default TRUE).

Value

Invisibly returns the trimmed+weighted data frame.

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
)

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` *Perform propensity score matching and optionally generate diagnostic reports.*

Description

This function performs propensity-score matching using **MatchIt** and generates comprehensive diagnostic reports including assumption checks, balance tables, and plots. The matching algorithm is selected via `method` (nearest / optimal / full / subclass).

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 the input data with PS already calculated (optional if in_csvpath provided)
in_csvpath	Character string. Path to input CSV file with PS already calculated (optional if in_df provided)
out_csvpath_matcheddata	Character string. Path for matched cohort CSV file (optional)
out_csvpath_crudedata_w_matchindicator	Character string. Path for crude data with match indicator CSV file (optional)
out_xlsxpath_report	Character string. Path for output Excel diagnostic report (optional).
out_dir_plots	Character string. Directory to save plot files (optional).
exposure_var	Character string. Name of the binary exposure/treatment variable column (default: "exp")
exp_value	Value representing the exposed/treated group (default: 1)
ref_value	Value representing the reference/control group (default: 0)

`ps_var` Character string. Name of the PS variable column in the data (default: "ps")

`method` Character. **MatchIt** matching method: "nearest" (default; the previous behavior) or "subclass". "subclass" does not produce 1:k pairs but returns matching weights (and a subclass) in a `.match_weights` column; for this method the matched Table 1 and any downstream effect estimation must be weighted by `.match_weights`.

`ratio` Integer. Matching ratio (1:k) for "nearest". Default 1.

`min_controls, max_controls`
Numeric or NULL. Variable-ratio bounds passed to **MatchIt** (`min.controls` / `max.controls`) for "nearest". `NULL` (default) reproduces fixed 1:`ratio`.

`m_order` Character or NULL. Matching order passed to **MatchIt** (`m.order`) for "nearest". `NULL` (default) keeps the MatchIt default.

`subclass_n` Integer or NULL. Number of subclasses when `method = "subclass"`. `NULL` (default) uses the MatchIt default.

`caliper` Numeric. Caliper width on the scale set by `caliper_scale`. Default is 0.2 (i.e. 0.2 x SD of logit(PS) under the default scale - the Austin (2011) convention).

`caliper_scale` Character. Scale on which matching and the caliper are applied. One of:

- `"logit_ps_sd"` (default): match on logit(PS); caliper = `caliper` x SD(logit(PS)). Requires PS strictly within (0, 1).
- `"raw_ps_sd"`: match on PS; caliper = `caliper` x SD(PS). This is the behavior of rwetools <= 0.1.x.
- `"raw"`: match on PS; flat caliper of `caliper` on the raw PS scale (e.g. `caliper = 0.01`).

`replace` Logical. Whether to match with replacement (method = "nearest"). Default FALSE.

`trim_method` Character. PS trimming applied before matching: "none" (default), "crump" (symmetric) or "sturmer" (asymmetric, exposure-group percentile tails). Trimming changes the analytic population and the interpretation of the estimand.

`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_crude_matched_table1`
Logical. Whether to generate crude and matched Table 1 (default: FALSE).

`table1_cont_vars`
Character vector. Names of continuous variables for Table 1 (optional, auto-detected if NULL)

`table1_binary_vars`
Character vector. Names of binary variables for Table 1 (optional, auto-detected if NULL)

`table1_cat_vars`
Character vector. Names of categorical variables for Table 1 (optional, auto-detected if NULL)

`std_diff_threshold`
Numeric. Threshold for acceptable standardized difference (default: 0.1)

`readme_text` Character string. Optional message to include in README sheet of Excel report

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

Value

A data frame containing the matched cohort only (crude data with match indicator can be saved to CSV via out_csvpath_crudedata_w_matchindicator).

Side Effects

- Writes matched-cohort and/or crude-data CSV files when the corresponding path arguments are provided.
- Creates directories, writes an Excel diagnostic report, and saves PNG plot files when the corresponding path arguments are supplied.

Examples

```
# Requires MatchIt
if (requireNamespace("MatchIt", quietly = TRUE)) {
  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
  )

  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_ir` *Estimate Incidence Rates, Hazard Ratios and Incidence Rate Ratios*

Description

Estimates incidence rates (IR), incidence rate differences (IRD), a hazard model (Cox HR or Fine-Gray subdistribution HR), and marginal incidence rate ratios (IRR) for a time-to-event outcome, for up to two analysis blocks: a crude block (`in_df_crude`) plus either a weighted block (`in_df_weight`, e.g. IPTW) or a matched block (`in_df_match`).

Usage

```
estimate_hr_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,
  stratification_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 the crude (unadjusted) block, or NULL.
`in_df_weight`	Data frame for the weighted block (requires `if_weight_weight_var`), or NULL. Cannot be combined with `in_df_match`.
`in_df_match`	Data frame for the matched block, or NULL. Cannot be combined with `in_df_weight`.
`out_xlsxpath`	Character path for an Excel output file, or NULL.
`exposure_var`	Character. Binary exposure variable name.
`exp_value, ref_value`	Values of `exposure_var` for the exposed and reference groups.
`outcome_var`	Character. Event indicator (0 = censored, 1 = event).
`followuptime_var`	Character. Follow-up time variable (required).
`time_unit`	"days", "months" or "years" – the unit of `followuptime_var`; person-years are derived as days/365.25, months/12, or years as-is.
`ir_per_pyears`	Multiplier for IR/IRD (1, 100, 1000, 10000, 100000).
`confidence_level`	Confidence level in (0, 1). Default 0.95.

`stratification_var`
: Character or NULL. When supplied, the hazard model is stratified via `strata()` and IR/IRD/IRR are direct-standardized with the total (weighted) person-time distribution as the standard.

`hr_model`
: "Cox" (cause-specific HR) or "Fine-Gray" (subdistribution HR; requires `if_fg_competing_event_var`).

`if_fg_competing_event_var`
: Character or NULL. Competing-event indicator (1 = competing event); required for `hr_model = "Fine-Gray"`, ignored (with a message) for "Cox".

`if_weight_weight_var`
: Character or NULL. Weight column in `in_df_weight`; required with that block, ignored (message) otherwise.

`if_match_match_id`
: Character or NULL. Match-set id column in `in_df_match`; enables pair clustering (HR/Fine-Gray SE) and pair-level bootstrap resampling. Without it the matched block uses a non-clustered robust SE and row resampling (a message is emitted). Ignored (message) without `in_df_match`.

`if_bootstrap_count`
: Integer or NULL. When supplied, adds percentile bootstrap CIs (columns `*_Boot`) for IR, IRD, IRR and the hazard estimate. The bootstrap is ALWAYS fixed-weight ("frozen"): weights and standardization shares are held at their original values, so PS-estimation uncertainty is not propagated (a message states this).

`if_bootstrap_n_cores`
: Integer or NULL. Cores for the bootstrap (NULL = all minus one); ignored (message) without `if_bootstrap_count`.

`if_bootstrap_seed`
: Integer or NULL. Bootstrap RNG seed; ignored (message) without `if_bootstrap_count`.

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

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

Details

Analytical confidence intervals are always computed with a fixed, per-block method (there are no CI-method arguments):

- Crude/matched IR: Garwood exact-Poisson (gamma); standardized IR arm CIs: Fay–Feuer gamma (single stratum collapses to Garwood).
- Weighted IR/IRD: design-based robust (sandwich) variance via a joint weighted quasi-Poisson cell model (Lumley 2004); standardized variants use the joint sandwich/delta method.
- HR: crude block uses the Cox model SE; weighted block the robust sandwich SE; matched block the robust SE clustered on `if_match_match_id` (Lin–Wei 1989; Austin 2016).
- Fine-Gray sHR (`hr_model = "Fine-Gray"`): robust SE clustered on the subject (crude/weighted) or the match id (matched); weighted expansion weights are IPCW x PS weight (Fine & Gray 1999).
- IRR (always reported): crude/matched blocks use a marginal Poisson rate model (model SE, equal to sqrt(1/D1 + 1/D0)); the weighted block uses survey::svyglm quasi-Poisson (robust SE). With `stratification_var`, the IRR is direct-standardized.

Value

Invisibly, a list with `incidence_rates` (one row set with per-block column suffixes `_Crude` / `_Weighted` / `_Matched`), `hazard_ratios` (Cox) or `subdist_hazard` (Fine-Gray), `incidence_rate_ratios`, per-block model objects (`models`), stratum details when standardized, bootstrap matrices when run, and `ir_per_pyears`.

Side Effects

Creates the output directory and writes an Excel workbook when `out_xlsxpath` is provided.

Examples

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

# Crude block only
res <- estimate_hr_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 + weighted blocks, Fine-Gray hazard model, 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_fg <- estimate_hr_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",
  hr_model                  = "Fine-Gray",
  if_fg_competing_event_var = "competing_event",
  verbose                   = FALSE
)
res_fg$subdist_hazard
```

```
res_boot <- estimate_hr_ir(
  in_df_crude          = df,
  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
)
```

`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).

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_rr_rd` *Estimate Risk Ratios and Risk Differences using Cumulative Incidence*

Description

Estimates risks (cumulative incidence), risk ratios (RR) and risk differences (RD) at a specified timepoint for up to two analysis blocks: a crude block (`in_df_crude`) plus either a weighted block (`in_df_weight`, e.g. IPTW) or a matched block (`in_df_match`). Two estimators are supported: Kaplan-Meier (`"KM"`, 1 - S(t)) and Aalen-Johansen (`"AJ"`, the cumulative incidence function under competing risks; requires `if_aj_competing_event_var`).

Usage

```
estimate_rr_rd(
  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"),
  rr_rd_at_timepoint = 365,
  risk_per_individuals = 1000,
  confidence_level = 0.95,
  risk_estimator = c("KM", "AJ"),
  if_aj_competing_event_var = NULL,
  stratification_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 the crude block, or NULL.
`in_df_weight`	Data frame for the weighted block (requires `if_weight_weight_var`); cannot be combined with `in_df_match`.
`in_df_match`	Data frame for the matched block; cannot be combined with `in_df_weight`.
`out_xlsxpath`	Character path for an Excel output file, or NULL.
`exposure_var`	Character. Binary exposure variable name.
`exp_value, ref_value`	Values of `exposure_var` for the exposed and reference groups.
`outcome_var`	Character. Event indicator (0 = censored, 1 = event).
`followuptime_var`	Character. Follow-up time variable (required).
`time_unit`	"days", "months" or "years" – the unit of `followuptime_var` and `rr_rd_at_timepoint`.
`rr_rd_at_timepoint`	Numeric timepoint for the cumulative incidence (same unit as `followuptime_var`). Default 365.
`risk_per_individuals`	Denominator for expressing risks and risk differences (default 1000).
`confidence_level`	Confidence level in (0, 1). Default 0.95.
`risk_estimator`	"KM" (Kaplan-Meier) or "AJ" (Aalen-Johansen; requires `if_aj_competing_event_var`). Aliases "Kaplan-Meier" / "Aalen-Johansen" are accepted.
`if_aj_competing_event_var`	Character or NULL. Competing-event indicator (1 = competing event) for the AJ estimator; required with `risk_estimator = "AJ"`, ignored (message) with "KM".
`stratification_var`	Character or NULL. Direct standardization variable (stratum weights from the total population).

`if_weight_weight_var`
: Character or NULL. Weight column in `in_df_weight`; required with that block, ignored (message) otherwise.

`if_match_match_id`
: Character or NULL. Match-set id column in `in_df_match`; enables pair-level bootstrap resampling. Without it the matched block uses row resampling (a message is emitted). Ignored (message) without `in_df_match`.

`if_bootstrap_count`
: Integer or NULL. When supplied, adds percentile bootstrap CIs (the `*_Boot` columns). Fixed-weight (frozen) only.

`if_bootstrap_n_cores`
: Integer or NULL. Cores for the bootstrap (NULL = all minus one); ignored (message) without `if_bootstrap_count`.

`if_bootstrap_seed`
: Integer or NULL. Bootstrap RNG seed; ignored (message) without `if_bootstrap_count`.

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

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

Details

Analytical confidence intervals are always computed with fixed methods:

- Risk / CIF: complementary log-log (cloglog) interval on the final (standardized/weighted) estimate with its combined SE (Kalbfleisch & Prentice 2002) – for a crude KM risk this reproduces `survfit(conf.type = "log-log")` exactly. A risk of exactly 0 or 1 has no defined cloglog CI: NA is returned and a message recommends the bootstrap.
- RD: normal-Wald with Greenwood (KM) / counting-process (AJ) variance. RR: delta method on the log scale.
- Standardization (with `stratification_var`): stratum weights w_k = N_k / N_total; `Var(Risk_std) = Sum(w_k^2 Var(R_k))`.

The percentile bootstrap is opt-in via `if_bootstrap_count` and is ALWAYS fixed-weight ("frozen"): analysis weights and stratum shares are held at their original values (PS-estimation uncertainty is not propagated); matched blocks resample matched sets by `if_match_match_id`.

Value

Invisibly, a list with `estimates` (one row per block; both analytical and `*_Boot` columns), `cumulative_incidence` (per-arm risks with cloglog CIs), `stratum_details` (when standardized), and per-block bootstrap matrices when run.

Side Effects

Creates the output directory and writes an Excel workbook when `out_xlsxpath` is provided.

Examples

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

# Crude KM risks with analytical (cloglog / Wald / log-delta) CIs
res <- estimate_rr_rd(
  in_df_crude          = df,
  exposure_var         = "exposure",
  outcome_var          = "outcome",
  followuptime_var     = "follow_up_days",
  time_unit            = "days",
  rr_rd_at_timepoint   = 365,
  risk_per_individuals = 1000,
  verbose              = FALSE
)
res$estimates


# AJ (competing risks) + bootstrap CIs
res2 <- estimate_rr_rd(
  in_df_crude               = df,
  exposure_var              = "exposure",
  outcome_var               = "outcome",
  followuptime_var          = "follow_up_days",
  rr_rd_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
)
res2$estimates
```

Index