Package 'PSsurvival'

January 10, 2026

Title Propensity Score Methods for Survival Analysis

Version 0.2.0

Date 2026-01-09

Description Implements propensity score weighting methods for estimating counterfactual survival functions, marginal hazard ratios, and weighted Kaplan-Meier and cumulative risk curves in observational studies with time-to-event outcomes. Supports binary and multiple treatment groups with inverse probability of treatment weighting (IPW), overlap weighting (OW), and average treatment effect on the treated (ATT). Includes symmetric trimming (Crump extension) for extreme propensity scores. Variance estimation via analytical M-estimation or bootstrap. Methods based on Li et al. (2018) <doi:10.1080/01621459.2016.1260466>, Li & Li (2019) <doi:10.1214/19-AOAS1282>, and Cheng et al. (2022) <doi:10.1093/aje/kwac043>.

License GPL (>= 2)

URL https://github.com/cxinyang/PSsurvival

BugReports https://github.com/cxinyang/PSsurvival/issues

Depends R (>= 3.5.0)

Imports survival, stats, utils, ggplot2, cowplot

Suggests nnet, parallel, testthat (>= 3.0.0), knitr, rmarkdown

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Chengxin Yang [aut, cre],
Chao Cheng [aut],
Fan Li [aut],
Fan Li [aut]

Maintainer Chengxin Yang <chengxin.yang@duke.edu>

Repository CRAN

Date/Publication 2026-01-10 15:10:18 UTC

Contents

estimate_ps . . . 2
estimate_weights . . . 4
marCoxph . . . 6
plot.surveff . . . 10
plot.weightedKM . . . 11
print.marCoxph . . . 13
print.surveff . . . 14
print.weightedKM . . . 14
simdata_bin . . . 15
simdata_multi . . . 16
summary.marCoxph . . . 17
summary.surveff . . . 18
summary.weightedKM . . . 19
surveff . . . 20
weightedKM . . . 24

Index **28**

`estimate_ps` *Propensity Score Estimation for PSsurvival Package*

Description

Functions for estimating propensity scores for binary and multiple treatment groups. Estimate Propensity Scores

Fits a propensity score model and extracts propensity scores for binary or multiple treatment groups. For binary treatments, uses binomial logistic regression. For multiple treatments (>2 levels), uses multinomial logistic regression to estimate generalized propensity scores.

Usage

```
estimate_ps(data, treatment_var, ps_formula, ps_control = list())
```

Arguments

`data`	A data.frame containing the analysis data (typically the cleaned data with complete cases).
`treatment_var`	A character string specifying the name of the treatment variable in `data`. Can be numeric, character, or factor with any coding (e.g., 0/1, 1/2, "Control"/"Treated"). Function assumes treatment has been validated for 2 or more levels.

`ps_formula` A formula object for the propensity score model, of the form `treatment ~ covariates`.

`ps_control` An optional list of control parameters to pass to the model fitting function (`glm` for binary treatment or `nnet::multinom` for multiple treatments). Default is an empty list.

Details

Propensity Score Definition: Returns P(Z = observed | X) for each individual, not P(Z=1|X) for all (as in Rosenbaum & Rubin 1983). This definition enables direct use in IPW and extends naturally to multiple treatments.

Binary Treatments (2 levels): Fits binomial logistic regression via `glm()`. Treatment is factorized with levels sorted by `sort()`: numerically for numeric, alphabetically for character, by factor level order for factor. Returns P(Z = observed | X).

Multiple Treatments (>2 levels): Fits multinomial logistic regression via `nnet::multinom()`. Returns P(Z = observed | X) for each individual from the generalized PS matrix.

Control Parameters (`ps_control`):

- Binary: `glm.control()` parameters (default: `epsilon=1e-08, maxit=25`)
- Multiple: `multinom()` parameters (default: `MaxNWts=10000, maxit=100, trace=FALSE`)

Value

A list with the following components:

`ps_model` The fitted propensity score model object (class `glm` for binary treatment or `multinom` for multiple treatments).

`ps` A numeric vector of propensity scores representing the probability of receiving the actual treatment each individual received. Length equals the number of rows in `data`.

`ps_matrix` A numeric matrix of dimension n × K where n is the number of observations and K is the number of treatment levels. Each row contains the predicted probabilities for all treatment levels. Column names correspond to treatment levels.

`n_levels` An integer indicating the number of treatment levels.

`treatment_levels` A vector of unique treatment values sorted by `sort()`: numerically for numeric, alphabetically for character, by factor level order for factor.

Examples

```
# Example 1: Binary treatment
data(simdata_bin)
ps_bin <- estimate_ps(
  data = simdata_bin,
  treatment_var = "Z",
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2
)
summary(ps_bin$ps)
```

```
table(simdata_bin$Z)

# Example 2: Multiple treatments
data(simdata_multi)
ps_multi <- estimate_ps(
  data = simdata_multi,
  treatment_var = "Z",
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2
)
head(ps_multi$ps_matrix)
```

`estimate_weights` *Estimate Propensity Score Weights*

Description

Calculates propensity score weights for causal inference with optional trimming. Supports ATE, ATT, and overlap population estimands for binary and multiple treatment groups.

Usage

```
estimate_weights(
  ps_result,
  data,
  treatment_var,
  estimand = "ATE",
  att_group = NULL,
  trim = NULL,
  delta = NULL,
  alpha = NULL
)
```

Arguments

`ps_result`	A list returned by `estimate_ps()`, containing the fitted propensity score model and estimated propensity scores.
`data`	A data.frame containing the treatment variable (same data used in `estimate_ps()`).
`treatment_var`	A character string specifying the name of the treatment variable in `data`.
`estimand`	Character string specifying the target population. One of: • "ATE": Average Treatment Effect (default). Uses IPW method. • "ATT": Average Treatment Effect on the Treated. Uses IPW method. • "overlap": Overlap population (Li & Li, 2019). Uses overlap weighting.
`att_group`	For ATT estimation, specifies which treatment group to target. This is MANDATORY when estimand = "ATT". Ignored for other estimands.

trim	Character string specifying the trimming method, or NULL for no trimming (default). Options: "symmetric" (Crump extension) or "asymmetric" (Sturmer extension). Trimming is NOT supported with overlap estimand.
delta	Trimming threshold for symmetric trimming in (0, 1/J], where J is the number of treatment levels. If NULL (default), uses recommended values from Yoshida et al. (2019). Ignored unless trim = "symmetric".
alpha	Percentile threshold for asymmetric trimming in (0, 0.5). If NULL (default), uses recommended values from Yoshida et al. (2019). Ignored unless trim = "asymmetric".

Details

Trimming Workflow: When trimming is requested, the function: (1) identifies observations to trim using PS from full data, (2) re-estimates PS on trimmed data, (3) calculates weights from re-estimated PS. This ensures trimming uses the original covariate distribution while weights reflect the overlapping population.

Overlap weights do not support trimming (already bounded in [0,1]).

Value

A list containing:

weights	Numeric vector of weights (length = nrow(data)).
trim_summary	Data frame with trimming summary by treatment group.
ess	Named numeric vector of effective sample size by treatment group.
method	Character string: "IPW" for ATE/ATT, "overlap" for overlap.
estimand	Character string of estimand used.
att_group	Target group for ATT (NULL if not applicable).
trim_method	Character string of trimming method (NULL if no trimming).
delta	Numeric trimming threshold used for symmetric trimming (NULL if not applicable).
alpha	Numeric percentile threshold used for asymmetric trimming (NULL if not applicable).
n_levels	Number of treatment levels.
treatment_levels	Vector of treatment level values.
ps_result	PS result object (refitted after trimming if trimming was applied).

References

Li, F., & Li, F. (2019). Propensity score weighting for causal inference with multiple treatments. *The Annals of Applied Statistics*, 13(4), 2389-2415.

Yoshida, K., et al. (2019). Multinomial extension of propensity score trimming methods: A simulation study. *American Journal of Epidemiology*, 188(3), 609-616.

Crump, R. K., et al. (2009). Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96(1), 187-199.

Examples

```
# Example 1: Overlap weighting for binary treatment
data(simdata_bin)
ps_bin <- estimate_ps(
  data = simdata_bin,
  treatment_var = "Z",
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2
)
weights_ow <- estimate_weights(
  ps_result = ps_bin,
  data = simdata_bin,
  treatment_var = "Z",
  estimand = "overlap"
)
summary(weights_ow$weights)

# Example 2: ATT with multiple treatments
data(simdata_multi)
ps_multi <- estimate_ps(
  data = simdata_multi,
  treatment_var = "Z",
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2
)
weights_att <- estimate_weights(
  ps_result = ps_multi,
  data = simdata_multi,
  treatment_var = "Z",
  estimand = "ATT",
  att_group = "C"
)
summary(weights_att$weights)
```

`marCoxph` *Marginal Cox Model with Propensity Score Weighting*

Description

Main user interface for estimating marginal hazard ratios using propensity score weighting. Supports binary and multiple treatment groups with various weighting schemes (IPW, OW, or ATT) and optional trimming. Variance can be estimated via bootstrap or robust sandwich estimator.

Usage

```
marCoxph(
  data,
  ps_formula,
  time_var,
```

```
  event_var,
  reference_level,
  weight_method = "IPW",
  att_group = NULL,
  trim = FALSE,
  delta = NULL,
  variance_method = "bootstrap",
  boot_level = "full",
  B = 100,
  parallel = FALSE,
  mc.cores = 2,
  seed = NULL,
  ps_control = list(),
  robust = TRUE
)
```

Arguments

`data`	Data frame containing treatment, survival outcome, and covariates.
`ps_formula`	Formula for propensity score model: `treatment ~ covariates`.
`time_var`	Character string specifying the time-to-event variable name.
`event_var`	Character string specifying the event indicator variable name. Should be coded as 1=event, 0=censored.
`reference_level`	Treatment level to use as reference in Cox model. MANDATORY. Must be one of the treatment levels.
`weight_method`	Weighting method: "IPW" (inverse probability weighting), "OW" (overlap weighting), or "ATT" (average treatment effect on the treated). Default "IPW".
`att_group`	Target group for ATT. Required if `weight_method = "ATT"`.
`trim`	Logical. Perform symmetric propensity score trimming? Default FALSE. If TRUE, symmetric trimming is applied (Crump extension for multiple treatments). See `estimate_weights` for trimming details. Ignored if `weight_method = "OW"`. Asymmetric trimming is no longer supported due to poor statistical performance.
`delta`	Threshold for symmetric trimming in $(0, 1/J]$, where J is the number of treatment levels. Default NULL uses recommended values: 0.1 for binary treatment, 0.067 for 3 groups, $1/(2J)$ for $J \geq 4$ (Yoshida et al., 2019). Used only if `trim = TRUE`.
`variance_method`	Variance estimation method: "bootstrap" (default) or "robust". "bootstrap" resamples the entire analysis pipeline. "robust" uses the sandwich variance estimator from `coxph()` without bootstrap.
`boot_level`	Bootstrap sampling level: "full" (default) or "strata". "full" resamples from entire dataset (standard for observational studies). "strata" resamples within each treatment group preserving group sizes (useful when treatment assignment follows a stratified or fixed-ratio design). Only used if `variance_method = "bootstrap"`.

B	Number of bootstrap iterations. Default 100. Used only if variance_method = "bootstrap".
parallel	Logical. Use parallel bootstrap computation? Default FALSE.
mc.cores	Number of cores for parallel bootstrap. Default 2.
seed	Random seed for bootstrap reproducibility. Default NULL.
ps_control	Control parameters for propensity score model. Default list().
robust	Logical. Use robust (sandwich) variance in Cox model fitting? Default TRUE. When TRUE, coxph() is called with robust = TRUE.

Details

Weighting Methods:

The weight_method parameter specifies the target population for causal inference:

- **IPW (Inverse Probability Weighting)**: Observations are weighted by the inverse probability of their observed treatment, $w_i = 1/e_j(X_i)$ where j is the observed treatment group. Inference targets the combined population (ATE type).
- **OW (Overlap Weighting)**: Observations are weighted by overlap weights, which extends to multiple treatment groups (Li et al., 2018; Li and Li, 2019). Inference targets the population at clinical equipoise (overlap population).
- **ATT (Average Treatment Effect on the Treated)**: IPW weights tilted toward a specified target group. Observations in the target group receive weight 1, others receive $w_i = e_{\text{target}}(X_i)/e_j(X_i)$. Inference targets the specified treatment group population (ATT type).

Analysis Workflow: 1. Extract treatment variable from ps_formula. 2. Estimate propensity scores using multinomial logistic regression (or logistic for binary treatment). 3. Calculate propensity score weights based on weight_method and optional trim. 4. Fit marginal Cox model Surv(time, event) ~ treatment with weights. 5. Estimate variance via bootstrap (resampling full pipeline) or robust sandwich estimator.

Variance Estimation: - bootstrap: Resamples data (full or stratified), re-estimates PS and weights, re-fits Cox model. Provides bootstrap SE for log hazard ratios. - robust: Uses robust sandwich variance from coxph() directly. No bootstrap performed (faster but may be less accurate with extreme weights).

Value

Object of class "marCoxph" containing:

coxph_fitted	Fitted coxph model object.
logHR_est	Named vector of estimated log hazard ratios. Names are formatted as "treatment_var:level" (e.g., "Z:B" for treatment Z, level B vs reference).
logHR_se_robust	Named vector of robust standard errors from coxph.
logHR_se_bootstrap	Named vector of bootstrap standard errors. NULL if variance_method = "robust".

`n_coxph_fitted` Named vector of sample sizes per treatment group used in Cox model fitting (after trimming).

`events_coxph_fitted` Named vector of event counts per treatment group used in Cox model fitting (after trimming).

`variance_method` Variance method used: "bootstrap-full", "bootstrap-strata", or "robust".

`estimand` Target estimand used.

`att_group` Target group for ATT (NULL if not applicable).

`trim_method` Trimming method (NULL if no trimming).

`delta` Symmetric trimming threshold (NULL if not applicable).

`alpha` Asymmetric trimming threshold (NULL if not applicable).

`treatment_var` Name of treatment variable.

`treatment_levels` Sorted unique treatment values.

`reference_level` Reference level used in Cox model.

`n_levels` Number of treatment groups.

`n` Number of complete cases used in analysis.

`ps_result` Propensity score estimation results.

`weight_result` Weight estimation results.

`boot_result` Bootstrap results (NULL if `variance_method = "robust"`). Contains: boot_samples, boot_allocation, n_success_by_group, B.

References

Li, F., Morgan, K. L., & Zaslavsky, A. M. (2018). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521), 390-400.

Li, F., & Li, F. (2019). Propensity score weighting for causal inference with multiple treatments. *The Annals of Applied Statistics*, 13(4), 2389-2415.

Yoshida, K., et al. (2019). Multinomial extension of propensity score trimming methods: A simulation study. *American Journal of Epidemiology*, 188(3), 609-616.

Examples

```
# Example 1: Binary treatment with overlap weighting
data(simdata_bin)
result1 <- marCoxph(
  data = simdata_bin,
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2,
  time_var = "time",
  event_var = "event",
  reference_level = "A",
  weight_method = "OW"
)
```

```
summary(result1)

# Example 2: Multiple treatments with ATT and robust variance
data(simdata_multi)
result2 <- marCoxph(
  data = simdata_multi,
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2,
  time_var = "time",
  event_var = "event",
  reference_level = "C",
  weight_method = "ATT",
  att_group = "C",
  variance_method = "robust"
)
summary(result2)
```

`plot.surveff` *Plot Method for surveff Objects*

Description

Plot Method for surveff Objects

Usage

```
## S3 method for class 'surveff'
plot(
  x,
  type = "surv",
  max_time = NULL,
  strata_to_plot = NULL,
  strata_colors = NULL,
  conf_level = 0.95,
  include_CI = TRUE,
  curve_width = 1,
  CI_alpha = 0.3,
  legend_position = "right",
  legend_title = NULL,
  plot_title = NULL,
  ...
)
```

Arguments

x	A `surveff` object.
type	Type of plot: "surv" for survival curves or "survdiff" for treatment effect curves. Default "surv".

max_time	Maximum time to display on x-axis. If NULL, uses max(eval_times).
strata_to_plot	Vector of strata to plot. For type = "surv", must be subset of treatment_levels. For type = "survdiff", must be subset of contrast names (column names of difference_estimates). If NULL, plots all available strata.
strata_colors	Vector of color names/codes for strata. Length must match strata_to_plot. Order matches strata order. If NULL, uses ggplot2 default colors.
conf_level	Confidence level for confidence intervals. Default 0.95.
include_CI	Logical. Include confidence interval ribbons? Default TRUE.
curve_width	Line width for survival/difference curves. Default 1.
CI_alpha	Transparency level for CI ribbons (0-1). Default 0.3.
legend_position	Position of legend: "right" or "bottom". Default "right".
legend_title	Title for legend. If NULL, uses "Treatment" for type="surv" or "Comparison" for type="survdiff".
plot_title	Plot title. If NULL, uses default title based on type.
...	Additional arguments (ignored).

Details

Creates publication-ready plots of survival curves or treatment effects over time.

For type = "surv": Plots estimated survival functions with optional confidence intervals. Y-axis ranges from 0 to 1.

For type = "survdiff": Plots estimated treatment effects (survival differences) with optional confidence intervals. Y-axis is not constrained to [0,1].

Value

A ggplot2 object.

---

plot.weightedKM *Plot Method for Weighted Kaplan-Meier Estimates*

Description

Plot Method for Weighted Kaplan-Meier Estimates

Usage

```
## S3 method for class 'weightedKM'
plot(
  x,
  type = "Kaplan-Meier",
  include_CI = TRUE,
  conf_type = "log-log",
```

```
  conf_level = 0.95,
  max_time = NULL,
  strata_to_plot = NULL,
  strata_colors = NULL,
  curve_width = 1,
  CI_alpha = 0.3,
  legend_position = "right",
  legend_title = NULL,
  plot_title = NULL,
  xlab = "Time",
  ylab = NULL,
  xlim = NULL,
  ylim = NULL,
  risk_table = FALSE,
  risk_table_stats = c("n.risk", "n.acc.event"),
  risk_table_height = 0.25,
  risk_table_breaks = NULL,
  risk_table_fontsize = 3.5,
  ...
)
```

Arguments

x	An object of class "weightedKM" from `weightedKM()`.
type	Type of curve to plot: "Kaplan-Meier" (survival probabilities, default) or "CR" (cumulative risk, aka. cumulative incidence = 1 - survival).
include_CI	Logical. Include confidence interval ribbons? Default TRUE.
conf_type	Type of confidence interval: "plain", "log", or "log-log" (default). See Details.
conf_level	Confidence level for intervals. Default 0.95.
max_time	Maximum time to display on x-axis. Default is maximum observed event time.
strata_to_plot	Character vector of treatment levels to plot. Default plots all groups.
strata_colors	Character vector of colors for each stratum in `strata_to_plot`. Must match length. Default uses ggplot2 colors.
curve_width	Line width for survival curves. Default 1.
CI_alpha	Transparency level for confidence interval ribbons (0-1). Default 0.3.
legend_position	Position of legend: "right" or "bottom". Default "right".
legend_title	Title for legend. Default "Treatment".
plot_title	Main plot title. Default depends on `type`.
xlab	X-axis label. Default "Time".
ylab	Y-axis label. Default depends on `type`.
xlim	Numeric vector of length 2 specifying x-axis limits. Default `c(0, max_time)`.
ylim	Numeric vector of length 2 specifying y-axis limits. Default `c(0, 1)`.

`risk_table` Logical. Add risk table below the plot? Default FALSE. Requires `cowplot` package if TRUE.

`risk_table_stats` Character vector specifying statistics to display in risk table. Options: "n.risk" (number at risk), "n.acc.event" (cumulative events). Default `c("n.risk", "n.acc.event")`.

`risk_table_height` Numeric. Relative height of risk table (0-1). Default 0.25.

`risk_table_breaks` Numeric vector of time points for risk table columns. If NULL (default), automatically determined based on `max_time`.

`risk_table_fontsize` Numeric. Font size for risk table text. Default 3.5.

`...` Additional arguments (currently unused).

Details

When `type = "CR"`, the function plots $1 - S(t)$ representing the probability of experiencing the event by time t. Variance is the same as for survival, but confidence intervals are calculated on the CR scale.

Value

A ggplot2 object if `risk_table = FALSE`, or a combined plot (cowplot object) if `risk_table = TRUE`.

`print.marCoxph` *Print Method for marCoxph Objects*

Description

Print Method for marCoxph Objects

Usage

```
## S3 method for class 'marCoxph'
print(x, max.len = 10, round.digits = 4, ...)
```

Arguments

`x` A `marCoxph` object.

`max.len` Maximum number of treatment comparisons to print. Default 10.

`round.digits` Number of digits for rounding displayed values. Default 4.

`...` Additional arguments (ignored).

Value

Invisibly returns the input object `x`.

print.surveff *Print Method for surveff Objects*

Description

Print Method for surveff Objects

Usage

```
## S3 method for class 'surveff'
print(x, max.len = 6, round.digits = 4, ...)
```

Arguments

x	A surveff object.
max.len	Maximum number of rows (time points) to print. Default 6.
round.digits	Number of digits for rounding displayed values. Default 4.
...	Additional arguments (ignored).

Value

Invisibly returns the input object x.

print.weightedKM *Print Method for Weighted Kaplan-Meier Estimates*

Description

Prints a summary of weighted Kaplan-Meier survival estimates.

Usage

```
## S3 method for class 'weightedKM'
print(x, print.digits = 3, print.rows = 10, ...)
```

Arguments

x	An object of class "weightedKM" from weightedKM().
print.digits	Number of decimal places for printed output. Default 3.
print.rows	Number of rows to print for each treatment group. Default 10.
...	Additional arguments (currently unused).

Value

Invisibly returns the input object x.

Examples

```
data(simdata_bin)
result <- weightedKM(
  data = simdata_bin,
  treatment_var = "Z",
  time_var = "time",
  event_var = "event",
  weight_method = "none"
)
print(result)
```

`simdata_bin` *Simulated Survival Data with Binary Treatment*

Description

A simulated dataset for demonstrating propensity score weighting methods in survival analysis with a binary treatment.

Usage

```
simdata_bin
```

Format

A data frame with 1000 observations and 8 variables:

X1 Continuous covariate (standard normal).

X2 Continuous covariate (standard normal).

X3 Continuous covariate (standard normal).

B1 Binary covariate (0/1).

B2 Binary covariate (0/1).

Z Treatment group: "A" or "B". Distribution is approximately 40:60.

time Observed follow-up time (event or censoring), range 0-20.

event Event indicator: 1 = event observed, 0 = censored.

Details

The data were generated with the following characteristics:

- Treatment assignment depends on X1, X2, and B1 via logistic model.
- Survival times follow Weibull distributions with group-specific scales (group A has better survival than group B).
- Censoring times follow an exponential distribution depending on X1 and B1.
- Administrative censoring occurs at time 20.
- Overall censoring rate is approximately 30

See Also

simdata_multi for a dataset with 4 treatment groups.

Examples

```
data(simdata_bin)
head(simdata_bin)
table(simdata_bin$Z)
```

---

simdata_multi *Simulated Survival Data with Multiple Treatments*

Description

A simulated dataset for demonstrating propensity score weighting methods in survival analysis with four treatment groups.

Usage

```
simdata_multi
```

Format

A data frame with 1000 observations and 8 variables:

X1 Continuous covariate (standard normal).

X2 Continuous covariate (standard normal).

X3 Continuous covariate (standard normal).

B1 Binary covariate (0/1).

B2 Binary covariate (0/1).

Z Treatment group: "A", "B", "C", or "D". Distribution is approximately 20:20:20:35.

time Observed follow-up time (event or censoring), range 0-20.

event Event indicator: 1 = event observed, 0 = censored.

Details

The data were generated with the following characteristics:

- Treatment assignment depends on X1, X2, X3, B1, and B2 via multinomial logistic model.
- Survival times follow Weibull distributions with group-specific scales. Survival ordering (best to worst): C > A > B > D.
- Censoring times follow an exponential distribution depending on X1 and B1.
- Administrative censoring occurs at time 20.
- Overall censoring rate is approximately 30

See Also

simdata_bin for a dataset with binary treatment.

Examples

```
data(simdata_multi)
head(simdata_multi)
table(simdata_multi$Z)
```

---

summary.marCoxph *Summary Method for marCoxph Objects*

Description

Summary Method for marCoxph Objects

Usage

```
## S3 method for class 'marCoxph'
summary(object, conf_level = 0.95, round.digits = 4, style = "prints", ...)
```

Arguments

object	A marCoxph object.
conf_level	Confidence level for intervals. Default 0.95.
round.digits	Number of digits for rounding displayed values. Default 4. Only used if style = "prints".
style	Output style: "prints" (print formatted tables) or "returns" (return vectors). Default "prints".
...	Additional arguments (ignored).

Details

Confidence intervals are Wald-type intervals calculated as:

- Log scale: logHR ± z_crit * SE
- Original scale: exp(logHR ± z_crit * SE)

The SE used depends on variance_method from the original marCoxph call:

- "robust": Uses logHR_se_robust from sandwich estimator.
- "bootstrap-full" or "bootstrap-strata": Uses logHR_se_bootstrap.

Value

If `style = "prints"`, returns invisibly. If `style = "returns"`, returns a list with:

`logHR`	Named vector of log hazard ratio estimates.
`logHR_CI_lower`	Named vector of lower CI bounds on log scale.
`logHR_CI_upper`	Named vector of upper CI bounds on log scale.
`SE`	Named vector of standard errors on log scale (from variance_method).
`HR`	Named vector of hazard ratio estimates (original scale).
`HR_CI_lower`	Named vector of lower CI bounds on original scale.
`HR_CI_upper`	Named vector of upper CI bounds on original scale.
`variance_method`	Variance method used.
`conf_level`	Confidence level used.
`n_per_group`	Named vector of sample sizes per group in Cox model.
`events_per_group`	Named vector of event counts per group in Cox model.

`summary.surveff` *Summary Method for surveff Objects*

Description

Summary Method for surveff Objects

Usage

```
## S3 method for class 'surveff'
summary(
  object,
  conf_level = 0.95,
  max.len = 6,
  round.digits = 4,
  style = "prints",
  ...
)
```

Arguments

`object`	A `surveff` object.
`conf_level`	Confidence level for intervals. Default 0.95.
`max.len`	Maximum number of rows (time points) to print. Default 6. Only used if `style = "prints"`.

round.digits	Number of digits for rounding displayed values. Default 4. Only used if style = "prints".
style	Output style: "prints" (print formatted tables) or "returns" (return list of matrices). Default "prints".
...	Additional arguments (ignored).

Value

If style = "prints", returns invisibly. If style = "returns", returns a list with:

survival_summary
: List of matrices, one per treatment group, with columns: Time, Estimate, SE, CI.lower, CI.upper

difference_summary
: List of matrices, one per contrast, with same columns. NULL if no contrasts estimated.

---

summary.weightedKM *Summary Method for Weighted Kaplan-Meier Estimates*

Description

Generates summary tables of weighted Kaplan-Meier survival or cumulative risk estimates with confidence intervals for each treatment group.

Usage

```
## S3 method for class 'weightedKM'
summary(
  object,
  type = "Kaplan-Meier",
  conf_type = "log-log",
  conf_level = 0.95,
  print.digits = 3,
  print.rows = 10,
  ...
)
```

Arguments

object	An object of class "weightedKM" from weightedKM().
type	Type of estimate to summarize: "Kaplan-Meier" (survival probabilities, default) or "CR" (cumulative risk, aka. cumulative incidence = 1 - survival).
conf_type	Type of confidence interval: "plain", "log", or "log-log" (default). See ?plot.weightedKM for details.
conf_level	Confidence level for intervals. Default 0.95.

print.digits	Number of decimal places for printed output. Default 3.
print.rows	Number of rows to print for each treatment group. Default 10.
...	Additional arguments (currently unused).

Details

This method provides tabular summaries of weighted Kaplan-Meier estimates with confidence intervals. It uses the same CI calculation methods as `plot.weightedKM()`.

When `type = "CR"`, the function transforms survival estimates to cumulative risk $1 - S$ and calculates confidence intervals on that scale.

The returned list contains full-precision matrices that can be used for further analysis. The printed output is rounded for readability.

Value

A list with one element per treatment group. Each element is a matrix with columns:

- `time`: Evaluation time points
- `estimate`: Survival probability or cumulative risk
- `se`: Standard error
- `CI_lower`: Lower confidence bound
- `CI_upper`: Upper confidence bound

The list is returned invisibly with full precision. Printed output is rounded to `print.digits` decimal places and shows first `print.rows` rows per group.

---

surveff *Survival Effect Estimation with Propensity Score Weighting*

Description

Main user interface for estimating counterfactual survival functions and treatment effects using propensity score weighting and inverse probability of censoring weighting. Supports binary and multiple treatment groups with various weighting schemes (IPW, OW, or ATT) and optional trimming.

Usage

```
surveff(
  data,
  ps_formula,
  censoring_formula,
  eval_times = NULL,
  weight_method = "IPW",
  att_group = NULL,
  trim = FALSE,
```

```
  delta = NULL,
  contrast_matrix = NULL,
  censoring_method = "weibull",
  variance_method = NULL,
  B = 100,
  parallel = FALSE,
  mc.cores = 2,
  seed = NULL,
  censoring_control = NULL,
  ties = "efron",
  ps_control = list(),
  boot_level = "full"
)
```

Arguments

`data`	Data frame containing treatment, outcome, and covariates.
`ps_formula`	Formula for propensity score model: `treatment ~ covariates`.
`censoring_formula`	Formula for censoring model: `Surv(time, event) ~ covariates`. Event should be coded as 1=event, 0=censored. Use `I(1-event)` if reversed.
`eval_times`	Numeric vector of time points for evaluation. If NULL (default), uses all unique event times.
`weight_method`	Weighting method: "IPW" (inverse probability weighting), "OW" (overlap weighting), or "ATT" (average treatment effect on the treated). Default "IPW".
`att_group`	Target group for ATT. Required if `weight_method = "ATT"`.
`trim`	Logical. Perform symmetric propensity score trimming? Default FALSE. If TRUE, symmetric trimming is applied (Crump extension for multiple treatments). See `estimate_weights` for trimming details. Ignored if `weight_method = "OW"`. Asymmetric trimming is no longer supported due to poor statistical performance.
`delta`	Threshold for symmetric trimming in $(0, 1/J]$, where J is the number of treatment levels. Default NULL uses recommended values: 0.1 for binary treatment, 0.067 for 3 groups, $1/(2J)$ for $J \geq 4$ (Yoshida et al., 2019). Used only if `trim = TRUE`.
`contrast_matrix`	Optional matrix for treatment differences in multiple group settings. Each row defines one contrast with exactly two non-zero elements: -1 and 1. Column names must match treatment levels. For binary treatment, always estimates second level minus first level (S1 - S0), ignoring this parameter.
`censoring_method`	Method for censoring score estimation: "weibull" or "cox". Default "weibull".
`variance_method`	Variance estimation method: "analytical" (binary treatment with Weibull censoring only) or "bootstrap". Default "analytical" for binary Weibull, "bootstrap" otherwise. Cox censoring always uses bootstrap.

B	Number of bootstrap iterations. Default 100. Used only if variance_method = "bootstrap".
parallel	Logical. Use parallel bootstrap computation? Default FALSE.
mc.cores	Number of cores for parallel bootstrap. Default 2.
seed	Random seed for bootstrap reproducibility. Default NULL.
censoring_control	Control parameters passed to censoring model fitting function. For Weibull: passed to survreg(), default list(maxiter = 350). For Cox: passed to coxph(), default list().
ties	Tie handling method for Cox models. Default "efron". Ignored for Weibull.
ps_control	Control parameters for propensity score model. Default list().
boot_level	Bootstrap sampling level: "full" (default) or "strata". "full" resamples from entire dataset (standard for observational studies). "strata" resamples within each treatment group preserving group sizes (useful when treatment assignment follows a stratified or fixed-ratio design). Only used if variance_method = "bootstrap".

Details

Weighting Methods:

The weight_method parameter specifies the target population for causal inference:

- **IPW (Inverse Probability Weighting)**: Observations are weighted by the inverse probability of their observed treatment, $w_i = 1/e_j(X_i)$ where j is the observed treatment group. Inference targets the combined population.
- **OW (Overlap Weighting)**: Observations are weighted by overlap weights, which extends to multiple treatment groups (Li et al., 2018; Li and Li, 2019). Inference targets the population at clinical equipoise (overlap population).
- **ATT (Average Treatment Effect on the Treated)**: IPW weights tilted toward a specified target group. Observations in the target group receive weight 1, others receive $w_i = e_{\text{target}}(X_i)/e_j(X_i)$. Inference targets the specified treatment group population.

Variance Estimation: - Analytical: Binary treatment with Weibull censoring only (M-estimation). - Bootstrap: All settings (resamples entire pipeline). - Cox: Always uses bootstrap.

Treatment Effects: - Binary: S1 - S0 (second level minus first). - Multiple groups: Requires contrast_matrix for pairwise comparisons.

Value

List containing:

survival_estimates	Matrix [time x J] of survival function estimates for each group.
survival_se	Matrix [time x J] of standard errors for survival functions.

`difference_estimates`
: Matrix [time x K] of treatment effect estimates. For binary treatment: single column with S1-S0. For multiple groups: contrasts from `contrast_matrix`, or NULL if not provided.

`difference_se`
: Matrix [time x K] of standard errors for treatment effects.

`eval_times`
: Time points evaluated.

`treatment_levels`
: Sorted unique treatment values.

`n_levels`
: Number of treatment groups.

`n`
: Sample size (complete cases after data validation).

`included`
: Logical vector [n] indicating inclusion in analysis. TRUE = included, FALSE = excluded due to trimming.

`estimand`
: Estimand used.

`censoring_method`
: Censoring method used.

`variance_method`
: Variance method used.

`contrast_matrix`
: Contrast matrix used (NULL if not applicable).

`ps_model`
: Fitted propensity score model (glm or multinom object).

`censoring_models`
: Named list of fitted censoring models by treatment group.

`weights`
: Numeric vector [n] of final weights (0 for trimmed observations).

`trim_summary`
: Data frame with trimming summary by treatment group.

`ess`
: Named numeric vector of effective sample size by treatment group.

`boot_result`
: Bootstrap results (NULL if analytical variance used).

References

Li, F., Morgan, K. L., & Zaslavsky, A. M. (2018). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521), 390-400.

Li, F., & Li, F. (2019). Propensity score weighting for causal inference with multiple treatments. *The Annals of Applied Statistics*, 13(4), 2389-2415.

Yoshida, K., et al. (2019). Multinomial extension of propensity score trimming methods: A simulation study. *American Journal of Epidemiology*, 188(3), 609-616.

Cheng, C., Li, F., Thomas, L. E., & Li, F. (2022). Addressing extreme propensity scores in estimating counterfactual survival functions via the overlap weights. *American Journal of Epidemiology*, 191(6), 1140-1151.

Examples

```
# Example 1: Binary treatment with overlap weighting and Weibull censoring model
data(simdata_bin)
result1 <- surveff(
```

```
  data = simdata_bin,
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2,
  censoring_formula = survival::Surv(time, event) ~ X1 + B1,
  weight_method = "OW",
  censoring_method = "weibull"
)
summary(result1)
plot(result1)

# Example 2: Multiple treatments with IPW and Cox censoring model
data(simdata_multi)
result2 <- surveff(
  data = simdata_multi,
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2,
  censoring_formula = survival::Surv(time, event) ~ X1 + B1,
  weight_method = "IPW",
  censoring_method = "cox",
  variance_method = "bootstrap",
  B = 100
)
summary(result2)
```

`weightedKM` *Weighted Kaplan-Meier Estimation with Propensity Score Weights*

Description

Computes weighted Kaplan-Meier survival or cumulative incidence curves using propensity score weights. Supports multiple treatment groups with various weighting schemes (IPW, OW, or ATT) and optional trimming. Special case `weight_method = "none"` provides classical (unweighted) Kaplan-Meier.

Usage

```
weightedKM(
  data,
  treatment_var,
  time_var,
  event_var,
  ps_formula = NULL,
  weight_method = "IPW",
  att_group = NULL,
  trim = FALSE,
  delta = NULL,
  ps_control = list()
)
```

Arguments

`data`	Data frame containing treatment, survival outcome, and covariates.
`treatment_var`	Character string specifying the name of the treatment variable.
`time_var`	Character string specifying the time-to-event variable name.
`event_var`	Character string specifying the event indicator variable name. Should be coded as 1=event, 0=censored.
`ps_formula`	Formula for propensity score model: `treatment ~ covariates`. Required unless `weight_method = "none"`.
`weight_method`	Weighting method: "IPW" (inverse probability weighting), "OW" (overlap weighting), "ATT" (average treatment effect on the treated), or "none" (unweighted). Default "IPW".
`att_group`	Target group for ATT. Required if `weight_method = "ATT"`.
`trim`	Logical. To perform symmetric propensity score trimming? Default FALSE. If TRUE, symmetric trimming is applied (Crump extension for multiple treatments). See `estimate_weights` for trimming details. Ignored if `weight_method = "none"` or `weight_method = "OW"`. Asymmetric trimming is no longer supported due to poor statistical performance.
`delta`	Threshold for symmetric trimming in $(0, 1/J]$, where J is the number of treatment levels. Default NULL uses recommended values: 0.1 for binary treatment, 0.067 for 3 groups, $1/(2J)$ for $J \geq 4$ (Yoshida et al., 2019). Used only if `trim = TRUE`.
`ps_control`	Control parameters for propensity score model. Default `list()`. Ignored if `weight_method = "none"`.

Details

Weighting Methods:

The `weight_method` parameter specifies the target population for causal inference:

- **IPW (Inverse Probability Weighting)**: Observations are weighted by the inverse probability of their observed treatment, $w_i = 1/e_j(X_i)$ where j is the observed treatment group. Inference targets the combined population.
- **OW (Overlap Weighting)**: Observations are weighted by overlap weights, which extends to multiple treatment groups (Li et al., 2018; Li and Li, 2019). Inference targets the population at clinical equipoise (overlap population).
- **ATT (Average Treatment Effect on the Treated)**: IPW weights tilted toward a specified target group. Observations in the target group receive weight 1, others receive $w_i = e_{\text{target}}(X_i)/e_j(X_i)$. Inference targets the specified treatment group population.
- **none**: No weighting applied (all weights = 1). Produces classical unweighted Kaplan-Meier estimates stratified by treatment.

Value

Object of class "weightedKM" containing:

`eval_times`	Numeric vector of all unique event times.
`surv_estimates`	Matrix [n_times x n_groups] of survival probability estimates.
`surv_var`	Matrix [n_times x n_groups] of variances.
`n_risk`	Matrix [n_times x n_groups] of weighted number at risk.
`n_event`	Matrix [n_times x n_groups] of weighted number of events.
`n_acc_event`	Matrix [n_times x n_groups] of cumulative weighted events up to each time.
`treatment_levels`	Sorted unique treatment values.
`weight_method`	Weighting method used.
`att_group`	Target group for ATT (NULL if not applicable).
`trim`	Logical indicating whether trimming was applied.
`delta`	Symmetric trimming threshold (NULL if trim = FALSE).
`n`	Number of complete cases used in analysis.
`ps_result`	Propensity score estimation results (NULL if `weight_method = "none"`).
`weight_result`	Weight estimation results (NULL if `weight_method = "none"`).
`weights`	Vector of weights used in estimation (all 1s if `weight_method = "none"`).

References

Li, F., Morgan, K. L., & Zaslavsky, A. M. (2018). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521), 390-400.

Li, F., & Li, F. (2019). Propensity score weighting for causal inference with multiple treatments. *The Annals of Applied Statistics*, 13(4), 2389-2415.

Yoshida, K., et al. (2019). Multinomial extension of propensity score trimming methods: A simulation study. *American Journal of Epidemiology*, 188(3), 609-616.

Examples

```
# Example 1: Classical (unweighted) Kaplan-Meier for binary treatment
data(simdata_bin)
result1 <- weightedKM(
  data = simdata_bin,
  treatment_var = "Z",
  time_var = "time",
  event_var = "event",
  weight_method = "none"
)
plot(result1)

# Example 2: Overlap-weighted Kaplan-Meier
result2 <- weightedKM(
  data = simdata_bin,
```

```
  treatment_var = "Z",
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2,
  time_var = "time",
  event_var = "event",
  weight_method = "OW"
)
summary(result2)

# Example 3: IPW-weighted Kaplan-Meier for multiple treatments
data(simdata_multi)
result3 <- weightedKM(
  data = simdata_multi,
  treatment_var = "Z",
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2,
  time_var = "time",
  event_var = "event",
  weight_method = "IPW"
)
plot(result3)

# Example 4: ATT with symmetric trimming
result4 <- weightedKM(
  data = simdata_multi,
  treatment_var = "Z",
  ps_formula = Z ~ X1 + X2 + X3 + B1 + B2,
  time_var = "time",
  event_var = "event",
  weight_method = "ATT",
  att_group = "A",
  trim = TRUE,
  delta = 0.1
)
summary(result4)
```

Index

∗ **datasets**
simdata_bin, 15
simdata_multi, 16

estimate_ps, 2
estimate_weights, 4, *7*, *21*, *25*

marCoxph, 6

plot.surveff, 10
plot.weightedKM, 11
print.marCoxph, 13
print.surveff, 14
print.weightedKM, 14

simdata_bin, 15, *17*
simdata_multi, *16*, 16
summary.marCoxph, 17
summary.surveff, 18
summary.weightedKM, 19
surveff, 20

weightedKM, 24