

Package: landest (via r-universe)

October 14, 2026

Type Package

Title Landmark Estimation of Survival and Treatment Effect

Version 1.2

Date 2023-08-25

Author Layla Parast

Maintainer Layla Parast <parast@austin.utexas.edu>

Description Provides functions to estimate survival and a treatment effect using a landmark estimation approach.

License GPL

Imports stats, survival

NeedsCompilation no

Repository <https://laylaparast.r-universe.dev>

Date/Publication 2023-08-26 01:25:04 UTC

RemoteUrl <https://github.com/cran/landest>

RemoteRef HEAD

RemoteSha 5e53cf9eeabda8e99d3cf085ee2201dfdb2593d2

Contents

landest-package	2
delta.iptw.km	3
delta.km	5
delta.land.obs	8
delta.land.rct	11
example_obs	14
example_rct	15
ps.wgt.fun	16
surv.iptw.km	17
surv.km	18
surv.land.obs	20
surv.land.rct	21

Index	24
--------------	-----------

Description

Provides functions to estimate the probability of survival past some specified time and the treatment effect, defined as the difference in survival at the specified time, using Kaplan-Meier estimation, landmark estimation for a randomized trial setting, inverse probability of treatment weighted (IPTW) Kaplan-Meier estimation, and landmark estimation for an observational study setting. The landmark estimation approaches provide improved efficiency by incorporating intermediate event information and are robust to model misspecification. The IPTW Kaplan-Meier approach and landmark estimation in an observational study setting approach account for potential selection bias.

Author(s)

Layla Parast

References

- Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282), 457-481.
- Xie, J., & Liu, C. (2005). Adjusted Kaplan-Meier estimator and log-rank test with inverse probability of treatment weighting for survival data. *Statistics in Medicine*, 24(20), 3089-3110.
- Parast, L., Tian, L., & Cai, T. (2014). Landmark Estimation of Survival and Treatment Effect in a Randomized Clinical Trial. *Journal of the American Statistical Association*, 109(505), 384-394.
- Parast, L. & Griffin B.A. (2017). Landmark Estimation of Survival and Treatment Effects in Observational Studies. *Lifetime Data Analysis*, 23:161-182.

Examples

```
data(example_rct)
delta.km(tl=example_rct$TL, dl = example_rct$DL, treat = example_rct$treat, tt=2)
#executable but takes time
#delta.land.rct(tl=example_rct$TL, dl = example_rct$DL, treat = example_rct$treat, tt=2,
#landmark = 1, short = cbind(example_rct$TS,example_rct$DS), z.cov = as.matrix(example_rct$Z))
data(example_obs)
delta.iptw.km(tl=example_obs$TL, dl = example_obs$DL, treat = example_obs$treat, tt=2,
cov.for.ps = as.matrix(example_obs$Z))
#executable but takes time
#delta.land.obs(tl=example_obs$TL, dl = example_obs$DL, treat = example_obs$treat, tt=2,
#landmark = 1, short = cbind(example_obs$TS,example_obs$DS), z.cov = as.matrix(example_obs$Z),
#cov.for.ps = as.matrix(example_obs$Z))
```

delta.iptw.km	<i>Estimates survival and treatment effect using inverse probability of treatment weighted (IPTW) Kaplan-Meier estimation</i>
---------------	---

Description

Estimates the probability of survival past some specified time and the treatment effect, defined as the difference in survival at the specified time, using inverse probability of treatment weighted (IPTW) Kaplan-Meier estimation

Usage

```
delta.iptw.km(tl, dl, treat, tt, var = FALSE, conf.int = FALSE, ps.weights = NULL,
weight.perturb = NULL, perturb.ps = FALSE, cov.for.ps = NULL)
```

Arguments

tl	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
dl	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
treat	treatment indicator, should be 0/1.
tt	the time of interest, function estimates the probability of survival past this time
var	TRUE or FALSE; indicates whether variance estimates for the treatment effect and survival in each group are requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether 95% confidence intervals for the treatment effect and survival in each group are requested, default is FALSE.
ps.weights	propensity score (or inverse probability of treatment) weights
weight.perturb	a $(n1+n0)$ by x matrix of weights where $n1$ = length of tl for treatment group 1 and $n0$ = length of tl for treatment group 0; used for perturbation-resampling, default is null. If var or conf.int is TRUE and weight.perturb is not provided, the function generates $\text{expnential}(1)$ weights.
perturb.ps	TRUE or FALSE indicating whether the weight.perturb matrix includes the perturbed propensity score (or inverse probability of treatment) weights; if cov.for.ps is supplied instead of ps.weights, this is forced to be TRUE.
cov.for.ps	matrix of covariates to be used to estimate propensity score (or inverse probability of treatment) weights; either ps.weights or cov.for.ps must be supplied.

Details

Let T_{Li} denote the time of the primary event of interest for person i , C_i denote the censoring time, Z_i denote the vector of baseline (pretreatment) covariates, and G_i be the treatment group indicator such that $G_i = 1$ indicates treatment and $G_i = 0$ indicates control. Due to censoring, we observe $X_{Li} = \min(T_{Li}, C_i)$ and $\delta_{Li} = I(T_{Li} \leq C_i)$. This function estimates survival at time t within

each treatment group, $S_j(t) = P(T_L > t | G = j)$ for $j = 1, 0$ and the treatment effect defined as $\Delta(t) = S_1(t) - S_0(t)$.

The inverse probability of treatment weighted (IPTW) Kaplan-Meier (KM) estimate of survival at time t for each treatment group is

$$\hat{S}_{IPTW, KM, j}(t) = \prod_{t_{kj} \leq t} \left[1 - \frac{d_{kj}^w}{y_{kj}^w} \right] \text{ if } t \geq t_{1j}, \text{ or } 1 \text{ if } t < t_{1j}$$

where $t_{1j}, \dots, t_{Dj}$ are the distinct observed event times of the primary outcome in treatment group j , $d_{kj}^w = \sum_{i: X_{Li}=t_{kj}, \delta_{Li}=1} \hat{W}_j(Z_i)^{-1} \delta_{Li} I(G_i = j)$ and $y_{kj}^w = \sum_{i: X_{Li} \geq t_{kj}} \hat{W}_j(Z_i)^{-1} I(G_i = j)$, $W_j(Z_i) = P(G_i = j | Z_i)$, and $\hat{W}_j(Z_i)$ is the estimated propensity score (see `ps.wgt.fun` for more information). The IPTW KM estimate of treatment effect at time t is $\hat{\Delta}_{IPTW, KM}(t) = \hat{S}_{IPTW, KM, 1}(t) - \hat{S}_{IPTW, KM, 0}(t)$.

To obtain variance estimates and construct confidence intervals, we use a perturbation-resampling method. Specifically, let $\{V^{(b)} = (V_1^{(b)}, \dots, V_n^{(b)})^T, b = 1, \dots, B\}$ be $n \times B$ independent copies of a positive random variable U from a known distribution with unit mean and unit variance such as an $\text{Exp}(1)$ distribution. To estimate the variance of our estimates, we appropriately weight the estimates using these perturbation weights to obtain perturbed values: $\hat{S}_{IPTW, KM, 0}(t)^{(b)}$, $\hat{S}_{IPTW, KM, 1}(t)^{(b)}$, and $\hat{\Delta}_{IPTW, KM}(t)^{(b)}$, $b = 1, \dots, B$. We then estimate the variance of each estimate as the empirical variance of the perturbed quantities. To construct confidence intervals, one can either use the empirical percentiles of the perturbed samples or a normal approximation.

Value

A list is returned:

<code>S.estimate.1</code>	the estimate of survival at the time of interest for treatment group 1, $\hat{S}_1(t) = P(T > t G = 1)$
<code>S.estimate.0</code>	the estimate of survival at the time of interest for treatment group 0, $\hat{S}_0(t) = P(T > t G = 0)$
<code>delta.estimate</code>	the estimate of treatment effect at the time of interest
<code>S.var.1</code>	the variance estimate of $\hat{S}_1(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>S.var.0</code>	the variance estimate of $\hat{S}_0(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>delta.var</code>	the variance estimate of $\hat{\Delta}(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>p.value</code>	the p-value from testing $\Delta(t) = 0$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>conf.int.normal.S.1</code>	a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.normal.S.0</code>	a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.normal.delta</code>	a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>

`conf.int.quantile.S.1`
 a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on sample quantiles of the perturbed values, described above; if `conf.int = TRUE`

`conf.int.quantile.S.0`
 a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on sample quantiles of the perturbed values, described above; if `conf.int = TRUE`

`conf.int.quantile.delta`
 a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on sample quantiles of the perturbed values, described above; if `conf.int = TRUE`

Author(s)

Layla Parast

References

Xie, J., & Liu, C. (2005). Adjusted Kaplan-Meier estimator and log-rank test with inverse probability of treatment weighting for survival data. *Statistics in Medicine*, 24(20), 3089-3110.

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1), 41-55.

Rosenbaum, P. R., & Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association*, 79(387), 516-524.

Examples

```
data(example_obs)
W.weight = ps.wgt.fun(treat = example_obs$treat, cov.for.ps = as.matrix(example_obs$Z))
delta.ipw.km(tl=example_obs$TL, dl = example_obs$DL, treat = example_obs$treat, tt=2,
ps.weights = W.weight)
delta.ipw.km(tl=example_obs$TL, dl = example_obs$DL, treat = example_obs$treat, tt=2,
cov.for.ps = as.matrix(example_obs$Z))
```

delta.km	<i>Estimates survival and treatment effect using Kaplan-Meier estimation</i>
----------	--

Description

Estimates the probability of survival past some specified time and the treatment effect, defined as the difference in survival at the specified time, using Kaplan-Meier estimation

Usage

```
delta.km(tl, dl, treat, tt, var = FALSE, conf.int = FALSE, weight.perturb = NULL)
```

Arguments

t1	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
d1	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
treat	treatment indicator, should be 0/1.
tt	the time of interest, function estimates the probability of survival past this time
var	TRUE or FALSE; indicates whether variance estimates for the treatment effect and survival in each group are requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether 95% confidence intervals for the treatment effect and survival in each group are requested, default is FALSE.
weight.perturb	a $(n1+n0)$ by x matrix of weights where $n1$ = length of t1 for treatment group 1 and $n0$ = length of t1 for treatment group 0; used for perturbation-resampling, default is null. If var or conf.int is TRUE and weight.perturb is not provided, the function generates exponential(1) weights.

Details

Let T_{Li} denote the time of the primary event of interest for person i , C_i denote the censoring time and G_i be the treatment group indicator such that $G_i = 1$ indicates treatment and $G_i = 0$ indicates control. Due to censoring, we observe $X_{Li} = \min(T_{Li}, C_i)$ and $\delta_{Li} = I(T_{Li} \leq C_i)$. This function estimates survival at time t within each treatment group, $S_j(t) = P(T_L > t | G = j)$ for $j = 1, 0$ and the treatment effect defined as $\Delta(t) = S_1(t) - S_0(t)$.

The Kaplan-Meier (KM) estimate of survival at time t for each treatment group is

$$\hat{S}_{KM,j}(t) = \prod_{t_{kj} \leq t} \left[1 - \frac{d_{kj}}{y_{kj}} \right] \text{ if } t \geq t_{1j}, \text{ or } 1 \text{ if } t < t_{1j}$$

where $t_{1j}, \dots, t_{Dj}$ are the distinct observed event times of the primary outcome in treatment group j , d_{kj} is the number of events at time t_{kj} in treatment group j , and y_{kj} is the number of patients at risk at t_{kj} in treatment group j . The Kaplan-Meier (KM) estimate of treatment effect at time t is $\hat{\Delta}_{KM}(t) = \hat{S}_{KM,1}(t) - \hat{S}_{KM,0}(t)$.

To obtain variance estimates and construct confidence intervals, we use a perturbation-resampling method. Specifically, let $\{V^{(b)} = (V_1^{(b)}, \dots, V_n^{(b)})^T, b = 1, \dots, B\}$ be $n \times B$ independent copies of a positive random variable U from a known distribution with unit mean and unit variance such as an $\text{Exp}(1)$ distribution. To estimate the variance of our estimates, we appropriately weight the estimates using these perturbation weights to obtain perturbed values: $\hat{S}_{KM,0}(t)^{(b)}$, $\hat{S}_{KM,1}(t)^{(b)}$, and $\hat{\Delta}_{KM}(t)^{(b)}$, $b = 1, \dots, B$. We then estimate the variance of each estimate as the empirical variance of the perturbed quantities. To construct confidence intervals, one can either use the empirical percentiles of the perturbed samples or a normal approximation.

Value

A list is returned:

S.estimate.1	the estimate of survival at the time of interest for treatment group 1, $\hat{S}_1(t) = P(T > t G = 1)$
--------------	---

<code>S.estimate.0</code>	the estimate of survival at the time of interest for treatment group 0, $\hat{S}_0(t) = P(T > t G = 0)$
<code>delta.estimate</code>	the estimate of treatment effect at the time of interest
<code>S.var.1</code>	the variance estimate of $\hat{S}_1(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>S.var.0</code>	the variance estimate of $\hat{S}_0(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>delta.var</code>	the variance estimate of $\hat{\Delta}(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>p.value</code>	the p-value from testing $\Delta(t) = 0$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>conf.int.normal.S.1</code>	a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.normal.S.0</code>	a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.normal.delta</code>	a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.quantile.S.1</code>	a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on sample quantiles of the perturbed values, described above; if <code>conf.int = TRUE</code>
<code>conf.int.quantile.S.0</code>	a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on sample quantiles of the perturbed values, described above; if <code>conf.int = TRUE</code>
<code>conf.int.quantile.delta</code>	a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on sample quantiles of the perturbed values, described above; if <code>conf.int = TRUE</code>

Author(s)

Layla Parast

References

Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282), 457-481.

Examples

```
data(example_rct)
delta.km(tl=example_rct$TL, dl = example_rct$DL, treat = example_rct$treat, tt=2)
```

delta.land.obs

*Estimates survival and treatment effect using landmark estimation***Description**

Estimates the probability of survival past some specified time and the treatment effect, defined as the difference in survival at the specified time, using landmark estimation for an observational study setting

Usage

```
delta.land.obs(tl, dl, treat, tt, landmark, short = NULL, z.cov = NULL,
var = FALSE, conf.int = FALSE, ps.weights = NULL, weight.perturb = NULL,
perturb.ps = FALSE, cov.for.ps = NULL, bw = NULL)
```

Arguments

tl	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
dl	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
treat	treatment indicator, should be 0/1.
tt	the time of interest, function estimates the probability of survival past this time
landmark	the landmark time
short	a matrix of intermediate event information, there should be two columns for each intermediate event, the first column contains the observed intermediate event time, equal to $\min(TS, C)$ where TS is the event time and C is the censoring time, and the second column contains the event indicator, equal to $I(TS < C)$
z.cov	matrix of baseline covariate information
var	TRUE or FALSE; indicates whether variance estimates for the treatment effect and survival in each group are requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether 95% confidence intervals for the treatment effect and survival in each group are requested, default is FALSE.
ps.weights	propensity score (or inverse probability of treatment) weights
weight.perturb	a $(n1+n0)$ by x matrix of weights where $n1$ = length of tl for treatment group 1 and $n0$ = length of tl for treatment group 0; used for perturbation-resampling, default is null. If var or conf.int is TRUE and weight.perturb is not provided, the function generates $\text{exponential}(1)$ weights.
perturb.ps	TRUE or FALSE indicating whether the weight.perturb matrix includes the perturbed propensity score (or inverse probability of treatment) weights; if cov.for.ps is supplied instead of ps.weights, this is forced to be TRUE.
cov.for.ps	matrix of covariates to be used to estimate propensity score (or inverse probability of treatment) weights; either ps.weights or cov.for.ps must be supplied.
bw	bandwidth used for kernel estimation, default is NULL

Details

Let T_{Li} denote the time of the primary event of interest for person i , T_{Si} denote the time of the available intermediate event(s), C_i denote the censoring time, Z_i denote the vector of baseline (pretreatment) covariates, and G_i be the treatment group indicator such that $G_i = 1$ indicates treatment and $G_i = 0$ indicates control. Due to censoring, we observe $X_{Li} = \min(T_{Li}, C_i)$ and $\delta_{Li} = I(T_{Li} \leq C_i)$ and $X_{Si} = \min(T_{Si}, C_i)$ and $\delta_{Si} = I(T_{Si} \leq C_i)$. This function estimates survival at time t within each treatment group, $S_j(t) = P(T_L > t | G = j)$ for $j = 1, 0$ and the treatment effect defined as $\Delta(t) = S_1(t) - S_0(t)$.

To derive these estimates using landmark estimation for an observational study setting, we first decompose the quantity into two components $S_j(t) = S_j(t|t_0)S_j(t_0)$ using a landmark time t_0 and estimate each component separately incorporating inverse probability of treatment weights (IPTW) to account for potential selection bias. Let $W_j(Z_i) = P(G_i = j | Z_i)$, and $\hat{W}_j(Z_i)$ be the estimated propensity score (or probability of treatment, see ps.wgt.fun for more information). In this presentation, we assume Z_i indicates the vector of baseline (pretreatment) covariates and that Z_i is used to estimate the propensity scores and incorporated into the survival and treatment effect estimation. However, the function allows one to use different subsets of Z_i for the propensity score estimation vs. survival estimation, as is appropriate in the setting of interest. Intermediate event information is used in estimation of the conditional component $S_j(t|t_0)$,

$$S_j(t|t_0) = P(T_L > t | T_L > t_0, G = j) = E[E[I(T_L > t | T_L > t_0, G = j, H)]] = E[S_{j,H}(t|t_0)]$$

where $S_{j,H}(t|t_0) = P(T_L > t | T_L > t_0, G = j, H)$ and $H = \{Z, I(T_S \leq t_0), \min(T_S, t_0)\}$. Then $S_{j,H}(t|t_0)$ is estimated in two stages. The first stage involves fitting a weighted Cox proportional hazards model among individuals with $X_L > t_0$ to obtain an estimate of β , denoted as $\hat{\beta}$,

$$S_{j,H}(t|t_0) = \exp\{-\Lambda_{j,0}(t|t_0) \exp(\beta^T H)\}$$

where $\Lambda_{j,0}(t|t_0)$ is the cumulative baseline hazard in group j . Specifically, $\hat{\beta}$ is the solution to the weighted Cox partial likelihood, with weights $\hat{W}_j(Z_i)^{-1}$. The second stage uses a weighted nonparametric kernel Nelson-Aalen estimator to obtain a local constant estimator for the conditional hazard $\Lambda_{j,u}(t|t_0) = -\log[S_{j,u}(t|t_0)]$ as

$$\hat{\Lambda}_{j,u}(t|t_0) = \int_{t_0}^t \frac{\sum_i \hat{W}_j(Z_i)^{-1} K_h(\hat{U}_i - u) dN_i(z)}{\sum_i \hat{W}_j(Z_i)^{-1} K_h(\hat{U}_i - u) Y_i(z)}$$

where $S_{j,u}(t|t_0) = P(T_L > t | T_L > t_0, G = j, \hat{U} = u)$, $\hat{U} = \hat{\beta}^T H$, $Y_i(t) = I(T_L \geq t)$, $N_i(t) = I(T_L \leq t)I(T_L < C)$, $K(\cdot)$ is a smooth symmetric density function, $K_h(x/h)/h$, $h = O(n^{-v})$ is a bandwidth with $1/2 > v > 1/4$, and the summation is over all individuals with $G = j$ and $X_L > t_0$. The resulting estimate for $S_{j,u}(t|t_0)$ is $\hat{S}_{j,u}(t|t_0) = \exp\{-\hat{\Lambda}_{j,u}(t|t_0)\}$, and the final estimate

$$\hat{S}_j(t|t_0) = \frac{n^{-1} \sum_{i=1}^n \hat{W}_j(Z_i)^{-1} \hat{S}_j(t|t_0, H_i) I(G_i = 1) I(X_{Li} > t_0)}{n^{-1} \sum_{i=1}^n \hat{W}_j(Z_i)^{-1} I(G_i = 1) I(X_{Li} > t_0)}$$

is a consistent estimate of $S_j(t|t_0)$.

Estimation of $S_j(t_0)$ uses a similar two-stage approach but using only baseline covariates, to obtain $\hat{S}_j(t_0)$. The final overall estimate of survival at time t is, $\hat{S}_{LM,j}(t) = \hat{S}_j(t|t_0)\hat{S}_j(t_0)$. The treatment effect in terms of the difference in survival at time t is estimated as $\hat{\Delta}_{LM}(t) = \hat{S}_{LM,1}(t) - \hat{S}_{LM,0}(t)$. To obtain an appropriate h we first use the bandwidth selection procedure given by Scott(1992) to obtain h_{opt} ; and then we let $h = h_{opt}n_j^{-0.10}$.

To obtain variance estimates and construct confidence intervals, we use a perturbation-resampling method. Specifically, let $\{V^{(b)} = (V_1^{(b)}, \dots, V_n^{(b)})^T, b = 1, \dots, B\}$ be $n \times B$ independent copies of a positive random variable U from a known distribution with unit mean and unit variance such as an $\text{Exp}(1)$ distribution. To estimate the variance of our estimates, we appropriately weight the estimates using these perturbation weights to obtain perturbed values: $\hat{S}_{LM,0}(t)^{(b)}$, $\hat{S}_{LM,1}(t)^{(b)}$, and $\hat{\Delta}_{LM}(t)^{(b)}$, $b = 1, \dots, B$. We then estimate the variance of each estimate as the empirical variance of the perturbed quantities. To construct confidence intervals, one can either use the empirical percentiles of the perturbed samples or a normal approximation.

Value

A list is returned:

S.estimate.1	the estimate of survival at the time of interest for treatment group 1, $\hat{S}_1(t) = P(T > t G = 1)$
S.estimate.0	the estimate of survival at the time of interest for treatment group 0, $\hat{S}_0(t) = P(T > t G = 0)$
delta.estimate	the estimate of treatment effect at the time of interest
S.var.1	the variance estimate of $\hat{S}_1(t)$; if var = TRUE or conf.int = TRUE
S.var.0	the variance estimate of $\hat{S}_0(t)$; if var = TRUE or conf.int = TRUE
delta.var	the variance estimate of $\hat{\Delta}(t)$; if var = TRUE or conf.int = TRUE
p.value	the p-value from testing $\Delta(t) = 0$; if var = TRUE or conf.int = TRUE
conf.int.normal.S.1	a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on a normal approximation; if conf.int = TRUE
conf.int.normal.S.0	a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on a normal approximation; if conf.int = TRUE
conf.int.normal.delta	a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on a normal approximation; if conf.int = TRUE
conf.int.quantile.S.1	a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE
conf.int.quantile.S.0	a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE
conf.int.quantile.delta	a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE

Author(s)

Layla Parast

References

- Parast, L. & Griffin B.A. (2017). Landmark Estimation of Survival and Treatment Effects in Observational Studies. *Lifetime Data Analysis*, 23:161-182.
- Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1), 41-55.
- Rosenbaum, P. R., & Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association*, 79(387), 516-524.

Examples

```
data(example_obs)
W.weight = ps.wgt.fun(treat = example_obs$treat, cov.for.ps = as.matrix(example_obs$Z))
#executable but takes time
#delta.land.obs(tl=example_obs$TL, dl = example_obs$DL, treat = example_obs$treat, tt=2,
#landmark = 1, short = cbind(example_obs$TS,example_obs$DS), z.cov = as.matrix(example_obs$Z),
#ps.weights = W.weight)
#delta.land.obs(tl=example_obs$TL, dl = example_obs$DL, treat = example_obs$treat, tt=2,
#landmark = 1, short = cbind(example_obs$TS,example_obs$DS), z.cov = as.matrix(example_obs$Z),
#cov.for.ps = as.matrix(example_obs$Z))
```

delta.land.rct

Estimates survival and treatment effect using landmark estimation

Description

Estimates the probability of survival past some specified time and the treatment effect, defined as the difference in survival at the specified time, using landmark estimation for a randomized trial setting

Usage

```
delta.land.rct(tl, dl, treat, tt, landmark, short = NULL, z.cov = NULL,
var = FALSE, conf.int = FALSE, weight.perturb = NULL, bw = NULL)
```

Arguments

tl	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
dl	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
treat	treatment indicator, should be 0/1.
tt	the time of interest, function estimates the probability of survival past this time
landmark	the landmark time

short	a matrix of intermediate event information, there should be two columns for each intermediate event, the first column contains the observed intermediate event time, equal to $\min(TS, C)$ where TS is the event time and C is the censoring time, and the second column contains the event indicator, equal to $I(TS < C)$
z.cov	matrix of baseline covariate information
var	TRUE or FALSE; indicates whether variance estimates for the treatment effect and survival in each group are requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether 95% confidence intervals for the treatment effect and survival in each group are requested, default is FALSE.
weight.perturb	a $(n1+n0)$ by x matrix of weights where $n1$ = length of tl for treatment group 1 and $n0$ = length of tl for treatment group 0; used for perturbation-resampling, default is null. If <code>var</code> or <code>conf.int</code> is TRUE and <code>weight.perturb</code> is not provided, the function generates exponential(1) weights.
bw	bandwidth used for kernel estimation, default is NULL

Details

Let T_{Li} denote the time of the primary event of interest for person i , T_{Si} denote the time of the available intermediate event(s), C_i denote the censoring time, Z_i denote the vector of baseline (pretreatment) covariates, and G_i be the treatment group indicator such that $G_i = 1$ indicates treatment and $G_i = 0$ indicates control. Due to censoring, we observe $X_{Li} = \min(T_{Li}, C_i)$ and $\delta_{Li} = I(T_{Li} \leq C_i)$ and $X_{Si} = \min(T_{Si}, C_i)$ and $\delta_{Si} = I(T_{Si} \leq C_i)$. This function estimates survival at time t within each treatment group, $S_j(t) = P(T_L > t | G = j)$ for $j = 1, 0$ and the treatment effect defined as $\Delta(t) = S_1(t) - S_0(t)$.

To derive these estimates using landmark estimation, we first decompose the quantity into two components $S_j(t) = S_j(t|t_0)S_j(t_0)$ using a landmark time t_0 and estimate each component separately. Intermediate event information is used in estimation of the conditional component $S_j(t|t_0)$,

$$S_j(t|t_0) = P(T_L > t | T_L > t_0, G = j) = E[E[I(T_L > t | T_L > t_0, G = j, H)]] = E[S_{j,H}(t|t_0)]$$

where $S_{j,H}(t|t_0) = P(T_L > t | T_L > t_0, G = j, H)$ and $H = \{Z, I(T_S \leq t_0), \min(T_S, t_0)\}$. Then $S_{j,H}(t|t_0)$ is estimated in two stages: 1) fitting the Cox proportional hazards model among individuals with $X_L > t_0$ to obtain an estimate of β , denoted as $\hat{\beta}$,

$$S_{j,H}(t|t_0) = \exp\{-\Lambda_{j,0}(t|t_0) \exp(\beta^T H)\}$$

where $\Lambda_{j,0}(t|t_0)$ is the cumulative baseline hazard in group j and then 2) using a nonparametric kernel Nelson-Aalen estimator to obtain a local constant estimator for the conditional hazard $\Lambda_{j,u}(t|t_0) = -\log[S_{j,u}(t|t_0)]$ as

$$\hat{\Lambda}_{j,u}(t|t_0) = \int_{t_0}^t \frac{\sum_i K_h(\hat{U}_i - u) dN_i(z)}{\sum_i K_h(\hat{U}_i - u) Y_i(z)}$$

where $S_{j,u}(t|t_0) = P(T_L > t | T_L > t_0, G = j, \hat{U} = u)$, $\hat{U} = \hat{\beta}^T H$, $Y_i(t) = I(T_L \geq t)$, $N_i(t) = I(T_L \leq t)I(T_L < C)$, $K(\cdot)$ is a smooth symmetric density function, $K_h(x/h)/h$, $h = O(n^{-v})$ is a bandwidth with $1/2 > v > 1/4$, and the summation is over all individuals with $G = j$ and

$X_L > t_0$. The resulting estimate for $S_{j,u}(t|t_0)$ is $\hat{S}_{j,u}(t|t_0) = \exp\{-\hat{\Lambda}_{j,u}(t|t_0)\}$, and the final estimate

$$\hat{S}_j(t|t_0) = \frac{n^{-1} \sum_{i=1}^n \hat{S}_j(t|t_0, H_i) I(G_i = 1) I(X_{Li} > t_0)}{n^{-1} \sum_{i=1}^n I(G_i = 1) I(X_{Li} > t_0)}$$

is a consistent estimate of $S_j(t|t_0)$.

Estimation of $S_j(t_0)$ uses a similar two-stage approach but using only baseline covariates, to obtain $\hat{S}_j(t_0)$. The final overall estimate of survival at time t is, $\hat{S}_{LM,j}(t) = \hat{S}_j(t|t_0) \hat{S}_j(t_0)$. The treatment effect in terms of the difference in survival at time t is estimated as $\hat{\Delta}_{LM}(t) = \hat{S}_{LM,1}(t) - \hat{S}_{LM,0}(t)$. To obtain an appropriate h we first use the bandwidth selection procedure given by Scott(1992) to obtain h_{opt} ; and then we let $h = h_{opt} n^{-0.10}$.

To obtain variance estimates and construct confidence intervals, we use a perturbation-resampling method. Specifically, let $\{V^{(b)} = (V_1^{(b)}, \dots, V_n^{(b)})^T, b = 1, \dots, B\}$ be $n \times B$ independent copies of a positive random variable U from a known distribution with unit mean and unit variance such as an $\text{Exp}(1)$ distribution. To estimate the variance of our estimates, we appropriately weight the estimates using these perturbation weights to obtain perturbed values: $\hat{S}_{LM,0}(t)^{(b)}$, $\hat{S}_{LM,1}(t)^{(b)}$, and $\hat{\Delta}_{LM}(t)^{(b)}$, $b = 1, \dots, B$. We then estimate the variance of each estimate as the empirical variance of the perturbed quantities. To construct confidence intervals, one can either use the empirical percentiles of the perturbed samples or a normal approximation.

Value

A list is returned:

S.estimate.1	the estimate of survival at the time of interest for treatment group 1, $\hat{S}_1(t) = P(T > t G = 1)$
S.estimate.0	the estimate of survival at the time of interest for treatment group 0, $\hat{S}_0(t) = P(T > t G = 0)$
delta.estimate	the estimate of treatment effect at the time of interest
S.var.1	the variance estimate of $\hat{S}_1(t)$; if var = TRUE or conf.int = TRUE
S.var.0	the variance estimate of $\hat{S}_0(t)$; if var = TRUE or conf.int = TRUE
delta.var	the variance estimate of $\hat{\Delta}(t)$; if var = TRUE or conf.int = TRUE
p.value	the p-value from testing $\Delta(t) = 0$; if var = TRUE or conf.int = TRUE
conf.int.normal.S.1	a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on a normal approximation; if conf.int = TRUE
conf.int.normal.S.0	a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on a normal approximation; if conf.int = TRUE
conf.int.normal.delta	a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on a normal approximation; if conf.int = TRUE
conf.int.quantile.S.1	a vector of size 2; the 95% confidence interval for $\hat{S}_1(t)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE

`conf.int.quantile.S.0`

a vector of size 2; the 95% confidence interval for $\hat{S}_0(t)$ based on sample quantiles of the perturbed values, described above; if `conf.int = TRUE`

`conf.int.quantile.delta`

a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on sample quantiles of the perturbed values, described above; if `conf.int = TRUE`

Author(s)

Layla Parast

References

Parast, L., Tian, L., & Cai, T. (2014). Landmark Estimation of Survival and Treatment Effect in a Randomized Clinical Trial. *Journal of the American Statistical Association*, 109(505), 384-394.

Beran, R. (1981). Nonparametric regression with randomly censored survival data. Technical report, University of California Berkeley.

Scott, D. (1992). Multivariate density estimation. Wiley.

Examples

```
data(example_rct)
#executable but takes time
#delta.land.rct(tl=example_rct$TL, dl = example_rct$DL, treat = example_rct$treat, tt=2,
#landmark = 1, short = cbind(example_rct$TS,example_rct$DS), z.cov = as.matrix(example_rct$Z))
```

example_obs

Hypothetical data from an observational study

Description

Hypothetical data from an observational study to be used in examples.

Usage

```
data(example_obs)
```

Format

A data frame with 4000 observations on the following 6 variables.

TL the observed event or censoring time for the primary outcome, equal to $\min(T, C)$ where T is the time of the primary outcome and C is the censoring time.

DL the indicator telling whether the individual was observed to have the event or was censored, equal to $1 \cdot (T < C)$ where T is the time of the primary outcome and C is the censoring time.

TS the observed event or censoring time for the intermediate event, equal to $\min(TS, C)$ where TS is the time of the intermediate event and C is the censoring time.

DS the indicator telling whether the individual was observed to have the intermediate event or was censored, equal to $1*(TS < C)$ where TS is the time of the primary outcome and C is the censoring time.

Z a baseline covariate vector

treat treatment indicator

Examples

```
data(example_obs)
names(example_obs)
```

example_rct	<i>Hypothetical data from a randomized trial</i>
-------------	--

Description

Hypothetical data from a randomized trial to be used in examples.

Usage

```
data(example_rct)
```

Format

A data frame with 3000 observations on the following 6 variables.

TL the observed event or censoring time for the primary outcome, equal to $\min(T, C)$ where T is the time of the primary outcome and C is the censoring time.

DL the indicator telling whether the individual was observed to have the event or was censored, equal to $1*(T < C)$ where T is the time of the primary outcome and C is the censoring time.

TS the observed event or censoring time for the intermediate event, equal to $\min(TS, C)$ where TS is the time of the intermediate event and C is the censoring time.

DS the indicator telling whether the individual was observed to have the intermediate event or was censored, equal to $1*(TS < C)$ where TS is the time of the primary outcome and C is the censoring time.

Z a baseline covariate vector

treat treatment indicator

Examples

```
data(example_rct)
names(example_rct)
```

ps.wgt.fun

Calculates propensity score weights

Description

Calculates propensity score (or inverse probability of treatment) weights given the treatment indicator and available baseline (pretreatment) covariates.

Usage

```
ps.wgt.fun(treat, cov.for.ps, weight = NULL)
```

Arguments

treat	treatment indicator, should be 0/1.
cov.for.ps	matrix of covariates to be used to estimate propensity score (or inverse probability of treatment) weights
weight	a (n1+n0) by x matrix of weights where n1 = number of observations in treatment group 1 and n0 = number of observations in treatment group 0; used for perturbation-resampling, default is null.

Details

Let Z_i denote the matrix of baseline (pretreatment) covariates and G_i be the treatment group indicator such that $G_i = 1$ indicates treatment and $G_i = 0$ indicates control. This function estimates $P = P(G_i = 1|Z_i)$ using logistic regression. The propensity score (or inverse probability of treatment) weights are then equal to $1/\hat{P}$ for those in treatment group 1 and $1/(1 - \hat{P})$ for those in treatment group 0. These weights reflect the situation where the average treatment effect (ATE) is of interest, not average treatment effect in the treated (ATT).

Value

propensity score (or inverse probability of treatment) weights

Author(s)

Layla Parast

References

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1), 41-55.

Rosenbaum, P. R., & Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association*, 79(387), 516-524.

Examples

```
data(example_obs)
W.weight = ps.wgt.fun(treat = example_obs$treat, cov.for.ps = as.matrix(example_obs$Z))
delta.iptw.km(tl=example_obs$TL, dl = example_obs$DL, treat = example_obs$treat, tt=2,
ps.weights = W.weight)
```

surv.iptw.km	<i>Estimates survival using inverse probability of treatment weighted (IPTW) Kaplan-Meier estimation</i>
--------------	--

Description

Estimates the probability of survival past some specified time using inverse probability of treatment weighted (IPTW) Kaplan-Meier estimation

Usage

```
surv.iptw.km(tl, dl, tt, var = FALSE, conf.int = FALSE, ps.weights,
weight.perturb = NULL, perturb.ps = FALSE, perturb.vector = FALSE)
```

Arguments

tl	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
dl	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
tt	the time of interest, function estimates the probability of survival past this time
var	TRUE or FALSE; indicates whether a variance estimate for survival is requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval for survival is requested, default is FALSE.
ps.weights	propensity score (or inverse probability of treatment) weights
weight.perturb	a n by x matrix of weights where n = length of tl; used for perturbation-resampling, default is null. If var or conf.int is TRUE and weight.perturb is not provided, the function generates $\text{exponential}(1)$ weights.
perturb.ps	TRUE or FALSE indicating whether the weight.perturb matrix includes the perturbed propensity score (or inverse probability of treatment) weights
perturb.vector	TRUE or FALSE; indicates whether a vector of the perturbed values of the survival estimate is requested, default is FALSE. This argument is ignored if both var and conf.int are FALSE.

Details

See documentation for delta.iptw.km for details.

Value

A list is returned:

<code>S.estimate</code>	the estimate of survival at the time of interest, $\hat{S}(t) = P(T > t)$
<code>S.var</code>	the variance estimate of $\hat{S}(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>conf.int.normal.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.quantile.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on sample quantiles of the perturbed values, described above; if <code>conf.int = TRUE</code>
<code>perturb.vector</code>	a vector of size x where x is the number of columns of the provided <code>weight.perturb</code> matrix (or x=500 if <code>weight.perturb</code> is not provided); the perturbed values of $\hat{S}(t)$; if <code>perturb.vector = TRUE</code> and either <code>var=TRUE</code> or <code>conf.int = TRUE</code>

Author(s)

Layla Parast

References

Xie, J., & Liu, C. (2005). Adjusted Kaplan-Meier estimator and log-rank test with inverse probability of treatment weighting for survival data. *Statistics in Medicine*, 24(20), 3089-3110.

Examples

```
data(example_obs)
W.weight = ps.wgt.fun(treat = example_obs$treat, cov.for.ps = as.matrix(example_obs$Z))
example_obs.treat = example_obs[example_obs$treat == 1,]
surv.ipw.km(tl=example_obs.treat$TL, dl = example_obs.treat$DL, tt=2, ps.weights =
W.weight[example_obs$treat == 1])
```

surv.km

Estimates survival using Kaplan-Meier estimation

Description

Estimates the probability of survival past some specified time using Kaplan-Meier estimation

Usage

```
surv.km(tl, dl, tt, var = FALSE, conf.int = FALSE, weight.perturb = NULL,
perturb.vector = FALSE)
```

Arguments

<code>tl</code>	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
<code>dl</code>	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
<code>tt</code>	the time of interest, function estimates the probability of survival past this time
<code>var</code>	TRUE or FALSE; indicates whether a variance estimate for survival is requested, default is FALSE.
<code>conf.int</code>	TRUE or FALSE; indicates whether a 95% confidence interval for survival is requested, default is FALSE.
<code>weight.perturb</code>	a n by x matrix of weights where $n = \text{length of } tl$; used for perturbation-resampling, default is null. If <code>var</code> or <code>conf.int</code> is TRUE and <code>weight.perturb</code> is not provided, the function generates $\text{exponential}(1)$ weights.
<code>perturb.vector</code>	TRUE or FALSE; indicates whether a vector of the perturbed values of the survival estimate is requested, default is FALSE. This argument is ignored if both <code>var</code> and <code>conf.int</code> are FALSE.

Details

See documentation for `delta.km` for details.

Value

A list is returned:

<code>S.estimate</code>	the estimate of survival at the time of interest, $\hat{S}(t) = P(T > t)$
<code>S.var</code>	the variance estimate of $\hat{S}(t)$; if <code>var</code> = TRUE or <code>conf.int</code> = TRUE
<code>conf.int.normal.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on a normal approximation; if <code>conf.int</code> = TRUE
<code>conf.int.quantile.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on sample quantiles of the perturbed values, described above; if <code>conf.int</code> = TRUE
<code>perturb.vector</code>	a vector of size x where x is the number of columns of the provided <code>weight.perturb</code> matrix (or $x=500$ if <code>weight.perturb</code> is not provided); the perturbed values of $\hat{S}(t)$; if <code>perturb.vector</code> = TRUE and either <code>var</code> =TRUE or <code>conf.int</code> = TRUE

Author(s)

Layla Parast

References

Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282), 457-481.

Examples

```
data(example_rct)
example_rct.treat = example_rct[example_rct$treat == 1,]
surv.km(tl=example_rct.treat$TL, dl = example_rct.treat$DL, tt=2)
```

surv.land.obs

Estimates survival using landmark estimation

Description

Estimates the probability of survival past some specified time using landmark estimation for an observational study setting

Usage

```
surv.land.obs(tl, dl, tt, landmark, short = NULL, z.cov = NULL, var = FALSE,
  conf.int = FALSE, ps.weights, weight.perturb = NULL, perturb.ps = FALSE,
  perturb.vector = FALSE, bw = NULL)
```

Arguments

tl	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
dl	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
tt	the time of interest, function estimates the probability of survival past this time
landmark	the landmark time
short	a matrix of intermediate event information, there should be two columns for each intermediate event, the first column contains the observed intermediate event time, equal to $\min(TS, C)$ where TS is the event time and C is the censoring time, and the second column contains the event indicator, equal to $I(TS < C)$
z.cov	matrix of baseline covariate information
var	TRUE or FALSE; indicates whether a variance estimate for survival is requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval for survival is requested, default is FALSE.
ps.weights	propensity score (or inverse probability of treatment) weights
weight.perturb	a n by x matrix of weights where n = length of tl; used for perturbation-resampling, default is null. If var or conf.int is TRUE and weight.perturb is not provided, the function generates $\text{exp}(1)$ weights.
perturb.ps	TRUE or FALSE indicating whether the weight.perturb matrix includes the perturbed propensity score (or inverse probability of treatment) weights
perturb.vector	TRUE or FALSE; indicates whether a vector of the perturbed values of the survival estimate is requested, default is FALSE. This argument is ignored if both var and conf.int are FALSE.
bw	bandwidth used for kernel estimation, default is NULL

Details

See documentation for delta.land.obs for details.

Value

A list is returned:

<code>S.estimate</code>	the estimate of survival at the time of interest, $\hat{S}(t) = P(T > t)$
<code>S.var</code>	the variance estimate of $\hat{S}(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>conf.int.normal.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.quantile.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on sample quantiles of the perturbed values, described above; if <code>conf.int = TRUE</code>
<code>perturb.vector</code>	a vector of size x where x is the number of columns of the provided weight.perturb matrix (or x=500 if weight.perturb is not provided); the perturbed values of $\hat{S}(t)$; if <code>perturb.vector = TRUE</code> and either <code>var=TRUE</code> or <code>conf.int = TRUE</code>

Author(s)

Layla Parast

References

Parast, L. & Griffin B.A. (2017). Landmark Estimation of Survival and Treatment Effects in Observational Studies. *Lifetime Data Analysis*, 23:161-182.

Examples

```
data(example_obs)
W.weight = ps.wgt.fun(treat = example_obs$treat, cov.for.ps = as.matrix(example_obs$Z))
example_obs.treat = example_obs[example_obs$treat == 1,]
#executable but takes time
#surv.land.obs(tl=example_obs.treat$TL, dl = example_obs.treat$DL, tt=2, landmark = 1,
#short = cbind(example_obs.treat$TS,example_obs.treat$DS), z.cov = example_obs.treat$Z,
#ps.weights = W.weight[example_obs$treat == 1])
```

surv.land.rct

Estimates survival using landmark estimation

Description

Estimates the probability of survival past some specified time using landmark estimation for a randomized trial setting

Usage

```
surv.land.rct(tl, dl, tt, landmark, short = NULL, z.cov = NULL, var = FALSE,
  conf.int = FALSE, weight.perturb = NULL, perturb.vector = FALSE, bw = NULL)
```

Arguments

<code>tl</code>	observed event time of primary outcome, equal to $\min(T, C)$ where T is the event time and C is the censoring time.
<code>dl</code>	event indicator, equal to $I(T < C)$ where T is the event time and C is the censoring time.
<code>tt</code>	the time of interest, function estimates the probability of survival past this time
<code>landmark</code>	the landmark time
<code>short</code>	a matrix of intermediate event information, there should be two columns for each intermediate event, the first column contains the observed intermediate event time, equal to $\min(TS, C)$ where TS is the event time and C is the censoring time, and the second column contains the event indicator, equal to $I(TS < C)$
<code>z.cov</code>	matrix of baseline covariate information
<code>var</code>	TRUE or FALSE; indicates whether a variance estimate for survival is requested, default is FALSE.
<code>conf.int</code>	TRUE or FALSE; indicates whether a 95% confidence interval for survival is requested, default is FALSE.
<code>weight.perturb</code>	a n by x matrix of weights where $n = \text{length of } tl$; used for perturbation-resampling, default is null. If <code>var</code> or <code>conf.int</code> is TRUE and <code>weight.perturb</code> is not provided, the function generates <code>expnential(1)</code> weights.
<code>perturb.vector</code>	TRUE or FALSE; indicates whether a vector of the perturbed values of the survival estimate is requested, default is FALSE. This argument is ignored if both <code>var</code> and <code>conf.int</code> are FALSE.
<code>bw</code>	bandwidth used for kernel estimation, default is NULL

Details

See documentation for `delta.land.rct` for details.

Value

A list is returned:

<code>S.estimate</code>	the estimate of survival at the time of interest, $\hat{S}(t) = P(T > t)$
<code>S.var</code>	the variance estimate of $\hat{S}(t)$; if <code>var = TRUE</code> or <code>conf.int = TRUE</code>
<code>conf.int.normal.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on a normal approximation; if <code>conf.int = TRUE</code>
<code>conf.int.quantile.S</code>	a vector of size 2; the 95% confidence interval for $\hat{S}(t)$ based on sample quantiles of the perturbed values, described above; if <code>conf.int = TRUE</code>

`perturb.vector` a vector of size `x` where `x` is the number of columns of the provided `weight.perturb` matrix (or `x=500` if `weight.perturb` is not provided); the perturbed values of $\hat{S}(t)$; if `perturb.vector = TRUE` and either `var=TRUE` or `conf.int = TRUE`

Author(s)

Layla Parast

References

Parast, L., Tian, L., & Cai, T. (2014). Landmark Estimation of Survival and Treatment Effect in a Randomized Clinical Trial. *Journal of the American Statistical Association*, 109(505), 384-394.

Examples

```
data(example_rct)
example_rct.treat = example_rct[example_rct$treat == 1,]
#executable but takes time
#surv.land.rct(tl=example_rct.treat$TL, dl = example_rct.treat$DL, tt=2, landmark = 1,
#short = cbind(example_rct.treat$TS,example_rct.treat$DS), z.cov = example_rct.treat$Z)
```

Index

- * **datasets**
 - example_obs, [14](#)
 - example_rct, [15](#)
 - * **nonlinear**
 - ps.wgt.fun, [16](#)
 - * **nonparametric**
 - delta.iptw.km, [3](#)
 - delta.km, [5](#)
 - delta.land.obs, [8](#)
 - delta.land.rct, [11](#)
 - surv.iptw.km, [17](#)
 - surv.km, [18](#)
 - surv.land.obs, [20](#)
 - surv.land.rct, [21](#)
 - * **package**
 - landest-package, [2](#)
 - * **regression**
 - ps.wgt.fun, [16](#)
 - * **robust**
 - delta.iptw.km, [3](#)
 - delta.land.obs, [8](#)
 - delta.land.rct, [11](#)
 - surv.iptw.km, [17](#)
 - surv.land.obs, [20](#)
 - surv.land.rct, [21](#)
 - * **survival**
 - delta.iptw.km, [3](#)
 - delta.km, [5](#)
 - delta.land.obs, [8](#)
 - delta.land.rct, [11](#)
 - landest-package, [2](#)
 - surv.iptw.km, [17](#)
 - surv.km, [18](#)
 - surv.land.obs, [20](#)
 - surv.land.rct, [21](#)
- delta.iptw.km, [3](#)
delta.km, [5](#)
delta.land.obs, [8](#)
delta.land.rct, [11](#)
- example_obs, [14](#)
example_rct, [15](#)
- landest (landest-package), [2](#)
landest-package, [2](#)
- ps.wgt.fun, [16](#)
- surv.iptw.km, [17](#)
surv.km, [18](#)
surv.land.obs, [20](#)
surv.land.rct, [21](#)