Package 'Rsurrogate'

July 21, 2025

Type Package

Title Robust Estimation of the Proportion of Treatment Effect Explained by Surrogate Marker Information

Version 3.2

Author Layla Parast

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

Description Provides functions to estimate the proportion of treatment effect on the primary outcome that is explained by the treatment effect on the surrogate marker.

License GPL

Imports stats, survival, Matrix

NeedsCompilation no

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2024-01-23 19:52:49 UTC

Contents

Aug.R.s.surv.estimate 2
delta.estimate 6
delta.multiple.surv 8
delta.s.estimate 10
delta.s.surv.estimate 12
delta.surv.estimate 13
delta.t.surv.estimate 16
d_example 17
d_example_me 18
d_example_multiple 18
d_example_surv 19
fieller.ci 20
me.variance.estimate 21
R.multiple.surv 22
R.s.estimate 28

R.s.estimate.me 33
R.s.surv.estimate 39
R.t.surv.estimate 44

Index **47**

`Aug.R.s.surv.estimate` *Calculates the augmented estimator of the proportion of treatment effect explained by the surrogate marker information measured at a specified time and primary outcome information up to that specified time*

Description

This function calculates the augmented version of the proportion of treatment effect on the primary outcome explained by the surrogate marker information measured at t_0 and primary outcome information up to t_0. Variance estimates and 95 % confidence intervals for the augmented estimates are provided automatically; three versions of the confidence interval are provided: a normal approximation based interval, a quantile based interval and Fieller's confidence interval, all using perturbation-resampling. The user can also request an estimate of the incremental value of surrogate marker information.

Usage

```
Aug.R.s.surv.estimate(xone, xzero, deltaone, deltazero, sone, szero, t,
weight.perturb = NULL, landmark, extrapolate = FALSE, transform = FALSE,
basis.delta.one, basis.delta.zero, basis.delta.s.one = NULL,
basis.delta.s.zero = NULL, incremental.value = FALSE, approx = T)
```

Arguments

`xone`	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`xzero`	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`deltaone`	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`deltazero`	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`sone`	numeric vector; surrogate marker measurement at t_0 for treated observations, assumed to be continuous. If $X_{1i} < t_0$, then the surrogate marker measurement should be NA.
`szero`	numeric vector; surrogate marker measurement at t_0 for control observations, assumed to be continuous. If $X_{1i} < t_0$, then the surrogate marker measurement should be NA.
`t`	the time of interest.

`weight.perturb` weights used for perturbation resampling.

`landmark` the landmark time t_0 or time of surrogate marker measurement.

`extrapolate` TRUE or FALSE; indicates whether the user wants to use extrapolation.

`transform` TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker.

`basis.delta.one`
either a vector of length n_1 or a matrix with n_1 rows; this is the basis transformation used for augmentation of $\hat{\Delta}(t)$ for treated observations only, all values must be numeric

`basis.delta.zero`
either a vector of length n_0 or a matrix with n_0 rows; this is the basis transformation used for augmentation of $\hat{\Delta}(t)$ for control observations only, all values must be numeric

`basis.delta.s.one`
either a vector of length n_1 or a matrix with n_1 rows; this is the basis transformation used for augmentation of $\hat{\Delta}_S(t, t_0)$ for treated observations only, all values must be numeric; default is to assume this is the same as basis.delta.one i.e. that the same basis transformation is used for both quantities

`basis.delta.s.zero`
either a vector of length n_0 or a matrix with n_0 rows; this is the basis transformation used for augmentation of $\hat{\Delta}_S(t, t_0)$ for control observations only, all values must be numeric; default is to assume this is the same as basis.delta.zero i.e. that the same basis transformation is used for both quantities

`incremental.value`
TRUE or FALSE; indicates whether the user would like to see the incremental value of the surrogate marker information, default is FALSE.

`approx` TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

Details

Please see R.s.surv.estimate documention for details about the estimates before augmentation is performed. Recent work has shown that augmentation can lead to improvements in efficiency by taking advantage of the association between baseline information, denoted here as Z, and the primary outcome. This function calculates the augmented estimates of the quantities of interest. For example, the augmented version of $\hat{\Delta}(t)$ is defined as:

$$\hat{\Delta}(t)^{AUG} = \hat{\Delta}(t) + \gamma\{n_1^{-1}\sum_{i=1}^{n_1} h(Z_{1i}) - n_0^{-1}\sum_{i=1}^{n_0} h(Z_{0i})\}$$

where $Z_{gi}, i = 1, 2, \cdots, n_g$ are i.i.d. random vectors of baseline covariates from treatment group g and $h(\cdot)$ is a basis transformation given a priori. Due to treatment randomization, $\{n_1^{-1}\sum_{i=1}^{n_1} h(Z_{1i}) - n_0^{-1}\sum_{i=1}^{n_0} h(Z_{0i})\}$ converges to zero in probability as the sample size goes to infinity and thus the

augmented estimator converges to the same limit as the original counterparts. The quantity γ is selected such that the variance of $\hat{\Delta}(t)^{AUG}$ is minimized. That is, $\gamma = (\Xi_{12})(\Xi_{22})^{-1}$ where

$$\Xi_{12} = \text{cov}\{\hat{\Delta}(t), n_1^{-1}\sum_{i=1}^{n_1} h(Z_{1i}) - n_0^{-1}\sum_{i=1}^{n_0} h(Z_{0i})\},$$

$$\Xi_{22} = \text{var}\{n_1^{-1}\sum_{i=1}^{n_1} h(Z_{1i}) - n_0^{-1}\sum_{i=1}^{n_0} h(Z_{0i})\}$$

and thus we can obtain $\hat{\Delta}(t)^{AUG}$ by replacing γ with a consistent estimator, $\hat{\gamma}$ obtained using perturbation-resampling. A similar approach is used to obtain $\hat{\Delta}_S(t)^{AUG}$ and thus construct

$$\hat{R}_S(t,t_0)^{AUG} = 1 - \frac{\hat{\Delta}_S(t,t_0)^{AUG}}{\hat{\Delta}(t)^{AUG}}.$$

When only a single Z_{gi} is provided in the basis argument, the following basis is used in this function: $h(Z_{gi}) = (1, Z_{gi}, Z_{gi}^2)'$.

Value

A list is returned:

`aug.delta` the estimate, $\hat{\Delta}(t)^{AUG}$.

`aug.delta.s` the estimate, $\hat{\Delta}_S(t,t_0)^{AUG}$.

`aug.R.s` the estimate, $\hat{R}_S(t,t_0)^{AUG}$.

`aug.delta.var` the variance estimate of $\hat{\Delta}(t)^{AUG}$.

`aug.delta.s.var` the variance estimate of $\hat{\Delta}_S(t,t_0)^{AUG}$.

`aug.R.s.var` the variance estimate of $\hat{R}_S(t,t_0)^{AUG}$.

`conf.int.normal.aug.delta` a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)^{AUG}$ based on a normal approximation.

`conf.int.quantile.aug.delta` a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)^{AUG}$ based on sample quantiles of the perturbed values.

`conf.int.normal.aug.delta.s` a vector of size 2; the 95% confidence interval for $\hat{\Delta}_S(t,t_0)^{AUG}$ based on a normal approximation.

`conf.int.quantile.aug.delta.s` a vector of size 2; the 95% confidence interval for $\hat{\Delta}_S(t,t_0)^{AUG}$ based on sample quantiles of the perturbed values.

`conf.int.normal.R.s` a vector of size 2; the 95% confidence interval for $\hat{R}_S(t,t_0)^{AUG}$ based on a normal approximation.

`conf.int.quantile.aug.R.s` a vector of size 2; the 95% confidence interval for $\hat{R}_S(t,t_0)^{AUG}$ based on sample quantiles of the perturbed values..

`conf.int.fieller.aug.R.s`
: a vector of size 2; the 95% confidence interval for $\hat{R}_S(t,t_0)^{AUG}$ based on Fieller's approach.

`aug.delta.t`
: the estimate, $\hat{\Delta}_T(t,t_0)^{AUG}$; if incremental.vaue = TRUE.

`aug.R.t`
: the estimate, $\hat{R}_T(t,t_0)^{AUG}$; if incremental.vaue = TRUE.

`aug.incremental.value`
: the estimate, $\hat{IV}_S(t,t_0)^{AUG}$; if incremental.vaue = TRUE.

`aug.delta.t.var`
: the variance estimate of $\hat{\Delta}_T(t,t_0)^{AUG}$; if incremental.vaue = TRUE.

`aug.R.t.var`
: the variance estimate of $\hat{R}_T(t,t_0)^{AUG}$; if incremental.vaue = TRUE.

`aug.incremental.value.var`
: the variance estimate of $\hat{IV}_S(t,t_0)^{AUG}$; if incremental.vaue = TRUE.

`aug.conf.int.normal.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t,t_0)^{AUG}$ based on a normal approximation; if incremental.vaue = TRUE.

`aug.conf.int.quantile.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t,t_0)^{AUG}$ based on sample quantiles of the perturbed values; if incremental.vaue = TRUE.

`aug.conf.int.normal.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t,t_0)^{AUG}$ based on a normal approximation; if incremental.vaue = TRUE.

`aug.conf.int.quantile.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t,t_0)^{AUG}$ based on sample quantiles of the perturbed values; if incremental.vaue = TRUE.

`aug.conf.int.fieller.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t,t_0)^{AUG}$ based on Fieller's approach, described above; if incremental.vaue = TRUE.

`aug.conf.int.normal.iv`
: a vector of size 2; the 95% confidence interval for $\hat{IV}_S(t,t_0)^{AUG}$ based on a normal approximation; if incremental.vaue = TRUE.

`aug.conf.int.quantile.iv`
: a vector of size 2; the 95% confidence interval for $\hat{IV}_S(t,t_0)^{AUG}$ based on sample quantiles of the perturbed values; if incremental.vaue = TRUE.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting". If the observed support of the surrogate marker for the control group is outside the observed support of the surrogate marker for the treatment group, the user will receive the following message: "Warning: observed supports do not appear equal, may need to consider a transformation or extrapolation".

Author(s)

Layla Parast

References

Tian L, Cai T, Zhao L,Wei L. On the covariate-adjusted estimation for an overall treatment difference with data from a randomized comparative clinical trial. Biostatistics 2012; 13(2): 256-273.

Garcia TP, Ma Y, Yin G. Efficiency improvement in a class of survival models through model-free covariate incorporation. Lifetime Data Analysis 2011; 17(4): 552-565.

Zhang M, Tsiatis AA, Davidian M. Improving efficiency of inferences in randomized clinical trials using auxiliary covariates. Biometrics 2008; 64(3): 707-715.

Parast, L., Cai, T., & Tian, L. (2017). Evaluating surrogate marker information using censored data. Statistics in Medicine, 36(11), 1767-1782.

Examples

```
#computationally intensive
#Aug.R.s.surv.estimate(xone = d_example_surv$x1, xzero = d_example_surv$x0,
#deltaone = d_example_surv$delta1, deltazero = d_example_surv$delta0,
#sone = d_example_surv$s1, szero = d_example_surv$s0, t=3, landmark = 1,
#basis.delta.one = d_example_surv$z1 , basis.delta.zero = d_example_surv$z0)
```

---

delta.estimate *Calculates treatment effect*

Description

This function calculates the treatment effect estimate, the difference in the average outcome in the treatment group minus the control group. This function is intended to be used for a fully observed continuous outcome. The user can also request a variance estimate, estimated using perturbating-resampling, and a 95% confidence interval. If a confidence interval is requested two versions are provided: a normal approximation based interval and a quantile based interval, both use perturbation-resampling.

Usage

```
delta.estimate(yone,yzero, var = FALSE, conf.int = FALSE, weight = NULL,
weight.perturb = NULL)
```

Arguments

yone	numeric vector; primary outcome for treated observations.
yzero	numeric vector; primary outcome for control observations.
var	TRUE or FALSE; indicates whether a variance estimate for delta is requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval for delta is requested, default is FALSE.

`weight` a n1+n0 by x matrix of weights where n1 = length of yone and n0 = length of yzero, default is null; generally not supplied by use but only used by other functions.

`weight.perturb` a n1+n0 by x matrix of weights where n1 = length of yone and n0 = length of yzero, default is null; generally used for confidence interval construction and may be supplied by user.

Details

Let $Y^{(1)}$ and $Y^{(0)}$ denote the primary outcome under the treatment and primary outcome under the control,respectively. The treatment effect, Δ, is the expected difference in $Y^{(1)}$ compared to $Y^{(0)}$, $\Delta = E(Y^{(1)} - Y^{(0)})$. We estimate Δ as

$$\hat{\Delta} = n_1^{-1} \sum_{i=1}^{n_1} Y_{1i} - n_0^{-1} \sum_{i=1}^{n_0} Y_{0i}$$

where Y_{1i} is the observed primary outcome for person i in the treated group, Y_{0i} is the observed primary outcome for person i in the control group, and n_1 and n_0 are the number of individuals in the treatment and control group, respectively. Randomized treatment assignment is assumed throughout this package.

Variance estimation and confidence interval construction are performed using perturbation-resampling. Specifically, let $\left\{ V^{(b)} = (V_{11}^{(b)}, ...V_{1n_1}^{(b)}, V_{01}^{(b)}, ...V_{0n_0}^{(b)})^T, b = 1, ..., D \right\}$ be $n \times D$ independent copies of a positive random variables V from a known distribution with unit mean and unit variance. Let

$$\hat{\Delta}^{(b)} = \frac{\sum_{i=1}^{n_1} V_{1i}^{(b)} Y_{1i}}{\sum_{i=1}^{n_1} V_{1i}^{(b)}} - \frac{\sum_{i=1}^{n_0} V_{0i}^{(b)} Y_{0i}}{\sum_{i=1}^{n_0} V_{0i}^{(b)}}.$$

The variance of $\hat{\Delta}$ is obtained as the empirical variance of $\{\hat{\Delta}^{(b)}, b = 1, ..., D\}$. In this package, we use weights generated from an Exponential(1) distribution and use $D = 500$. We construct two versions of the 95% confidence interval for $\hat{\Delta}$: one based on a normal approximation confidence interval using the estimated variance and another taking the 2.5th and 97.5th empirical percentiles of $\hat{\Delta}^{(b)}$.

Value

A list is returned:

`delta` the estimate, $\hat{\Delta}$, described above.

`var` the variance estimate of $\hat{\Delta}$; if var = TRUE or conf.int = TRUE.

`conf.int.normal` a vector of size 2; the 95% confidence interval for $\hat{\Delta}$ based on a normal approximation; if conf.int = TRUE.

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

Author(s)

Layla Parast

Examples

```
data(d_example)
names(d_example)
delta.estimate(yone=d_example$y1, yzero=d_example$y0)
```

`delta.multiple.surv` *Calculates robust residual treatment effect accounting for multiple surrogate markers at a specified time and primary outcome information up to that specified time*

Description

This function calculates the robust estimate of the residual treatment effect accounting for multiple surrogate markers measured at t_0 and primary outcome information up to t_0 i.e. the hypothetical treatment effect if both the surrogate marker distributions at t_0 and survival up to t_0 in the treatment group look like the surrogate marker distributions and survival up to t_0 in the control group. Ideally this function is only used as a helper function and is not directly called.

Usage

```
delta.multiple.surv(xone, xzero, deltaone, deltazero, sone, szero, type =1, t,
weight.perturb = NULL, landmark, extrapolate = FALSE, transform = FALSE,
approx = T)
```

Arguments

`xone`	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`xzero`	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`deltaone`	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`deltazero`	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`sone`	matrix of numeric values; surrogate marker measurements at t_0 for treated observations. If $X_{1i} < t_0$, then the surrogate marker measurements should be NA.
`szero`	matrix of numeric values; surrogate marker measurements at t_0 for control observations. If $X_{0i} < t_0$, then the surrogate marker measurements should be NA.
`type`	type of estimate; options are 1 = two-stage robust estimator, 2 = weighted two-stage robust estimator, 3 = double-robust estimator, 4 = two-stage model-based estimator, 5 = weighted estimator, 6 = double-robust model-bsed estimator; default is 1.
`t`	the time of interest.

weight.perturb	weights used for perturbation resampling.
landmark	the landmark time t_0 or time of surrogate marker measurement.
extrapolate	TRUE or FALSE; indicates whether the user wants to use extrapolation.
transform	TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker psuedo-score.
approx	TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

Details

Details are included in the documentation for R.multiple.surv.

Value

$\hat{\Delta}_S(t, t_0)$, the residual treatment effect estimate accounting for multiple surrogarte markers measured at t_0 and primary outcome information up to t_0.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting".

Author(s)

Layla Parast

References

Parast, L., Cai, T., & Tian, L. (2021). Evaluating multiple surrogate markers with censored data. Biometrics, 77(4), 1315-1327.

Examples

```
data(d_example_multiple)
names(d_example_multiple)
## Not run:
delta.multiple.surv(xone = d_example_multiple$x1, xzero = d_example_multiple$x0, deltaone =
 d_example_multiple$delta1, deltazero = d_example_multiple$delta0, sone =
 as.matrix(d_example_multiple$s1), szero = as.matrix(d_example_multiple$s0),
 type =1, t = 1, landmark=0.5)

## End(Not run)
```

`delta.s.estimate` *Calculates model-based or robust residual treatment effect*

Description

This function calculates the model-based or robust estimate of the residual treatment effect i.e. the hypothetical treatment effect if the distribution of the surrogate in the treatment group looks like the distribution of the surrogate in the control group. Ideally, this function is only used as a helper function and is not directly called.

Usage

```
delta.s.estimate(sone, szero, yone, yzero, weight.perturb = NULL, number="single",
type="robust", warn.te = FALSE, warn.support = FALSE, extrapolate = FALSE,
transform = FALSE)
```

Arguments

`sone`	numeric vector or matrix; surrogate marker for treated observations, assumed to be continuous. If there are multiple surrogates then this should be a matrix with n_1 (number of treated observations) rows and n.s (number of surrogate markers) columns.
`szero`	numeric vector or matrix; surrogate marker for control observations, assumed to be continuous. If there are multiple surrogates then this should be a matrix with n_0 (number of control observations) rows and n.s (number of surrogate markers) columns.
`yone`	numeric vector; primary outcome for treated observations.
`yzero`	numeric vector; primary outcome for control observations.
`weight.perturb`	a $n_1 + n_0$ by x matrix of weights where $n_1 =$ length of yone and $n_0 =$ length of yzero; generally used for variance estimation and confidence interval construction, default is null.
`number`	specifies the number of surrogate markers; choices are "multiple" or "single", default is "single".
`type`	specifies the type of estimation; choices are "robust" or "model", default is "robust".
`warn.te`	value passed from R.s.estimate function to control warnings; user does not need to specify.
`warn.support`	value passed from R.s.estimate function to control warnings; user does not need to specify.
`extrapolate`	TRUE or FALSE; indicates whether the user wants to use extrapolation.
`transform`	TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker.

Details

Details are included in the documentation for R.s.estimate.

Value

$\hat{\Delta}_S$, the model-based or robust residual treatment effect estimate.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting". In the single marker case with the robust estimation approach, if the observed support of the surrogate marker for the control group is outside the observed support of the surrogate marker for the treatment group, the user will receive the following message: "Warning: observed supports do not appear equal, may need to consider a transformation or extrapolation".

Author(s)

Layla Parast

References

Parast, L., McDermott, M., Tian, L. (2015). Robust estimation of the proportion of treatment effect explained by surrogate marker information. Statistics in Medicine, 35(10):1637-1653.

Wang, Y., & Taylor, J. M. (2002). A measure of the proportion of treatment effect explained by a surrogate marker. Biometrics, 58(4), 803-812.

Examples

```
data(d_example)
names(d_example)
delta.s.estimate(yone=d_example$y1, yzero=d_example$y0, sone=d_example$s1.a, szero=
d_example$s0.a, number = "single", type = "robust")
delta.s.estimate(yone=d_example$y1, yzero=d_example$y0, sone=d_example$s1.a, szero=
d_example$s0.a, number = "single", type = "model")
delta.s.estimate(yone=d_example$y1, yzero=d_example$y0, sone=cbind(d_example$s1.a,
d_example$s1.b, d_example$s1.c), szero=cbind(d_example$s0.a, d_example$s0.b, d_example$s0.c),
number = "multiple", type = "robust")
delta.s.estimate(yone=d_example$y1, yzero=d_example$y0, sone=cbind(d_example$s1.a,
d_example$s1.b, d_example$s1.c), szero=cbind(d_example$s0.a, d_example$s0.b, d_example$s0.c),
number = "multiple", type = "model")
```

`delta.s.surv.estimate` *Calculates robust residual treatment effect accounting for surrogate marker information measured at a specified time and primary outcome information up to that specified time*

Description

This function calculates the robust estimate of the residual treatment effect accounting for surrogate marker information measured at t_0 and primary outcome information up to t_0 i.e. the hypothetical treatment effect if both the surrogate marker distribution at t_0 and survival up to t_0 in the treatment group look like the surrogate marker distribution and survival up to t_0 in the control group. Ideally this function is only used as a helper function and is not directly called.

Usage

```
delta.s.surv.estimate(xone, xzero, deltaone, deltazero, sone, szero, t,
weight.perturb = NULL, landmark, extrapolate = FALSE, transform = FALSE,
approx = T, warn.te = FALSE, warn.support = FALSE)
```

Arguments

`xone`	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`xzero`	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`deltaone`	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`deltazero`	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`sone`	numeric vector; surrogate marker measurement at t_0 for treated observations, assumed to be continuous. If $X_{1i} < t_0$, then the surrogate marker measurement should be NA.
`szero`	numeric vector; surrogate marker measurement at t_0 for control observations, assumed to be continuous. If $X_{1i} < t_0$, then the surrogate marker measurement should be NA.
`t`	the time of interest.
`weight.perturb`	weights used for perturbation resampling.
`landmark`	the landmark time t_0 or time of surrogate marker measurement.
`extrapolate`	TRUE or FALSE; indicates whether the user wants to use extrapolation.
`transform`	TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker.
`approx`	TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

warn.te	value passed from R.s.estimate function to control warnings; user does not need to specify.
warn.support	value passed from R.s.estimate function to control warnings; user does not need to specify.

Details

Details are included in the documentation for R.s.surv.estimate.

Value

$\hat{\Delta}_S(t, t_0)$, the robust residual treatment effect estimate accounting for surrogate marker information measured at t_0 and primary outcome information up to t_0.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting". If the observed support of the surrogate marker for the control group is outside the observed support of the surrogate marker for the treatment group, the user will receive the following message: "Warning: observed supports do not appear equal, may need to consider a transformation or extrapolation".

Author(s)

Layla Parast

References

Parast, L., Cai, T., & Tian, L. (2017). Evaluating surrogate marker information using censored data. Statistics in Medicine, 36(11), 1767-1782.

Examples

```
data(d_example_surv)
names(d_example_surv)
```

---

delta.surv.estimate *Calculates treatment effect in a survival setting*

Description

This function calculates the treatment effect in the survival setting i.e. the difference in survival at time t between the treatment group and the control group. The user can also request a variance estimate, estimated using perturbating-resampling, and a 95% confidence interval. If a confidence interval is requested two versions are provided: a normal approximation based interval and a quantile based interval, both use perturbation-resampling.

Usage

```
delta.surv.estimate(xone, xzero, deltaone, deltazero, t, var = FALSE, conf.int
= FALSE, weight = NULL, weight.perturb = NULL, approx = T)
```

Arguments

xone	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
xzero	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
deltaone	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
deltazero	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
t	the time of interest.
var	TRUE or FALSE; indicates whether a variance estimate for delta is requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval for delta is requested, default is FALSE.
weight	a $n_1 + n_0$ by x matrix of weights where n_1 = sample size in treatment group and n_0 = sample size in the control group, default is null; generally not supplied by use but only used by other functions.
weight.perturb	a $n_1 + n_0$ by x matrix of weights where n_1 =sample size in treatment group and n_0 = sample size in the control group, default is null; generally used for confidence interval construction and may be supplied by user.
approx	TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

Details

Let G be the binary treatment indicator with $G = 1$ for treatment and $G = 0$ for control and we assume throughout that subjects are randomly assigned to a treatment group at baseline. Let T denote the time of the primary outcome of interest, death for example. We use potential outcomes notation such that $T^{(g)}$ denotes the time of the primary outcome under treatment $G = g$. We define the treatment effect, $\Delta(t)$, as the difference in survival rates by time t under treatment versus control,

$$\Delta(t) = E\{I(T^{(1)} > t)\} - E\{I(T^{(0)} > t)\} = P(T^{(1)} > t) - P(T^{(0)} > t)$$

where $t > t_0$

Due to censoring, our data consist of n_1 observations $\{(X_{1i}, \delta_{1i}), i = 1, ..., n_1\}$ from the treatment group $G = 1$ and n_0 observations $\{(X_{0i}, \delta_{0i}), i = 1, ..., n_0\}$ from the control group $G = 0$ where

$X_{gi} = \min(T_{gi}, C_{gi})$, $\delta_{gi} = I(T_{gi} < C_{gi})$, and C_{gi} denotes the censoring time for $g = 1, 0$, for individual i. Throughout, we estimate the treatment effect $\Delta(t)$ as

$$\hat{\Delta}(t) = n_1^{-1} \sum_{i=1}^{n_1} \frac{I(X_{1i} > t)}{\hat{W}_1^C(t)} - n_0^{-1} \sum_{i=1}^{n_0} \frac{I(X_{0i} > t)}{\hat{W}_0^C(t)}$$

where $\hat{W}_g^C(\cdot)$ is the Kaplan-Meier estimator of survival for censoring for $g = 1, 0$.

Variance estimation and confidence interval construction are performed using perturbation-resampling. Specifically, let $\left\{ V^{(b)} = (V_{11}^{(b)}, ...V_{1n_1}^{(b)}, V_{01}^{(b)}, ...V_{0n_0}^{(b)})^T, b = 1, ..., D \right\}$ be $n \times D$ independent copies of a positive random variables V from a known distribution with unit mean and unit variance. Let

$$\hat{\Delta}^{(b)}(t) = \frac{\sum_{i=1}^{n_1} V_{1i}^{(b)} I(X_{1i} > t)}{\sum_{i=1}^{n_1} V_{1i}^{(b)} \hat{W}_1^{C(b)}(t)} - \frac{\sum_{i=1}^{n_0} V_{0i}^{(b)} I(X_{0i} > t)}{\sum_{i=1}^{n_0} V_{0i}^{(b)} \hat{W}_0^{C(b)}(t)}.$$

In this package, we use weights generated from an Exponential(1) distribution and use $D = 500$. The variance of $\hat{\Delta}(t)$ is obtained as the empirical variance of $\{\hat{\Delta}(t)^{(b)}, b = 1, ..., D\}$. We construct two versions of the 95%confidence interval for $\hat{\Delta}(t)$: one based on a normal approximation confidence interval using the estimated variance and another taking the 2.5th and 97.5th empirical percentiles of $\hat{\Delta}(t)^{(b)}$.

Value

A list is returned:

`delta` the estimate, $\hat{\Delta}(t)$, described above.

`var` the variance estimate of $\hat{\Delta}(t)$; if var = TRUE or conf.int = TRUE.

`conf.int.normal` 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` 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

Examples

```
data(d_example_surv)
names(d_example_surv)
delta.surv.estimate(xone = d_example_surv$x1, xzero = d_example_surv$x0,
deltaone = d_example_surv$delta1, deltazero = d_example_surv$delta0, t = 3)
```

`delta.t.surv.estimate` *Calculates robust residual treatment effect accounting only for primary outcome information up to a specified time*

Description

This function calculates the robust estimate of the residual treatment effect accounting only for primary outcome information up to t_0 i.e. the hypothetical treatment effect if survival up to t_0 in the treatment group looks like survival up to t_0 in the control group. Ideally this function is only used as a helper function and is not directly called.

Usage

```
delta.t.surv.estimate(xone, xzero, deltaone, deltazero, t, weight.perturb = NULL,
landmark, approx = T)
```

Arguments

`xone`	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`xzero`	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`deltaone`	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`deltazero`	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`t`	the time of interest.
`weight.perturb`	weights used for perturbation resampling.
`landmark`	the landmark time t_0 or time of surrogate marker measurement.
`approx`	TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

Details

Details are included in the documentation for R.t.surv.estimate.

Value

$\hat{\Delta}_T(t, t_0)$, the robust residual treatment effect estimate accounting only for survival up to t_0.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting".

Author(s)

Layla Parast

References

Parast, L., Cai, T., & Tian, L. (2017). Evaluating surrogate marker information using censored data. Statistics in Medicine, 36(11), 1767-1782.

Examples

```
data(d_example_surv)
names(d_example_surv)
```

---

d_example *Hypothetical data*

Description

Hypothetical data to be used in examples.

Usage

```
data(d_example)
```

Format

A list with 8 elements representing 500 observations from a control group and 500 observations from a treatment group:

`s1.a` First surrogate marker measurement for treated observations.

`s1.b` Second surrogate marker measurement for treated observations.

`s1.c` Third surrogate marker measurement for treated observations.

`y1` Primary outcome for treated observations.

`s0.a` First surrogate marker measurement for control observations.

`s0.b` Second surrogate marker measurement for control observations.

`s0.c` Third surrogate marker measurement for control observations.

`y0` Primary outcome for control observations.

Examples

```
data(d_example)
names(d_example)
```

d_example_me *Hypothetical data with replicate measurements*

Description

Hypothetical data to be used in measurement error example.

Usage

```
data(d_example_me)
```

Format

A list with 10 elements representing 500 observations from a control group and 500 observations from a treatment group:

`y1` Primary outcome for treated observations.

`s1` Surrogate marker for treated observations.

`s1_rep1` Replicate measurement of the surrogate marker for treated observations.

`s1_rep2` Replicate measurement of the surrogate marker for treated observations.

`s1_rep3` Replicate measurement of the surrogate marker for treated observations.

`y0` Primary outcome for control observations.

`s0` Surrogate marker for control observations.

`s0_rep1` Replicate measurement of the surrogate marker for control observations.

`s0_rep2` Replicate measurement of the surrogate marker for control observations.

`s0_rep3` Replicate measurement of the surrogate marker for control observations.

Examples

```
data(d_example_me)
names(d_example_me)
```

d_example_multiple *Hypothetical survival data with multiple surrogate markers*

Description

Hypothetical survival data with multiple surrogate markers to be used in examples.

Usage

```
data(d_example_multiple)
```

Format

A list with 6 elements representing 1000 observations from a control group and 1000 observations from a treatment group:

`s1` Surrogate marker measurements for treated observations; these markers are measured at time = 0.5. For observations that experience the primary outcome or are censored before 0.5, the surrogate values are NA.

`x1` The observed event or censoring time for treated observations; X = min(T, C) where T is the time of the primary outcome and C is the censoring time.

`delta1` The indicator identifying whether the treated observation was observed to have the event or was censored; D =1*(T<C) where T is the time of the primary outcome and C is the censoring time.

`s0` Surrogate marker measurements for control observations; these markers are measured at time = 0.5. For observations that experience the primary outcome or are censored before 0.5, the surrogate values are NA.

`x0` The observed event or censoring time for control observations; X = min(T, C) where T is the time of the primary outcome and C is the censoring time.

`delta0` The indicator identifying whether the control observation was observed to have the event or was censored; D =1*(T<C) where T is the time of the primary outcome and C is the censoring time.

Examples

```
data(d_example_multiple)
names(d_example_multiple)
```

---

d_example_surv *Hypothetical survival data*

Description

Hypothetical survival data to be used in examples.

Usage

```
data(d_example_surv)
```

Format

A list with 8 elements representing 500 observations from a control group and 500 observations from a treatment group:

`s1` Surrogate marker measurement for treated observations; this marker is measured at time = 0.5. For observations that experience the primary outcome or are censored before 0.5, this value is NA.

`x1` The observed event or censoring time for treated observations; X = min(T, C) where T is the time of the primary outcome and C is the censoring time.

`delta1` The indicator identifying whether the treated observation was observed to have the event or was censored; D =1*(T<C) where T is the time of the primary outcome and C is the censoring time.

`s0` Surrogate marker measurement for control observations; this marker is measured at time = 0.5. For observations that experience the primary outcome or are censored before 0.5, this value is NA.

`x0` The observed event or censoring time for control observations; X = min(T, C) where T is the time of the primary outcome and C is the censoring time.

`delta0` The indicator identifying whether the control observation was observed to have the event or was censored; D =1*(T<C) where T is the time of the primary outcome and C is the censoring time.

`z1` A baseline covariate value for treated observations.

`z0` A baseline covariate value for control observations.

Examples

```
data(d_example_surv)
names(d_example_surv)
```

`fieller.ci` *Constructs Fieller's confidence interval.*

Description

Constructs Fieller's confidence interval.

Usage

```
fieller.ci(perturb.delta.s, perturb.delta, delta.s, delta)
```

Arguments

`perturb.delta.s`	numeric vector; the perturbed values for $\hat{\Delta}_S$, the residual treatment effect estimate, used in variance estimation and confidence interval construction.
`perturb.delta`	numeric vector; the perturbed values for $\hat{\Delta}$, the treatment effect estimate, used in variance estimation and confidence interval construction.
`delta.s`	the residual treatment effect, Δ_S, estimate, $\hat{\Delta}_S$.
`delta`	the treatment effect, Δ, estimate, $\hat{\Delta}$.

Details

See documention for R.s.estimate for more detail.

Value

Returns a vector of length 2, lower bound of the 95% confidence interval and upper bound of the 95% confidence interval.

Author(s)

Layla Parast

References

Fieller, Edgar C. (1954). Some problems in interval estimation. Journal of the Royal Statistical Society. Series B (Methodological), 175-185.

Fieller, E. C. (1940). The biological standardization of insulin. Supplement to the Journal of the Royal Statistical Society, 1-64.

Parast, L., McDermott, M., Tian, L. (2016). Robust estimation of the proportion of treatment effect explained by surrogate marker information. Statistics in Medicine, 35(10):1637-1653.

---

me.variance.estimate *Estimates measurement error variance given replicate data.*

Description

Estimates measurement error variance given replicate data using a simple components of variance analysis.

Usage

```
me.variance.estimate(replicates)
```

Arguments

`replicates`	matrix of data where each row indicates a subject and each column is a replicated measurement; columns can have NAs when subjects have different numbers of measurements.

Details

Estimates measurement error variance given replicate data using a simple components of variance analysis.

Value

estimate of measurement error variance

Author(s)

Layla Parast

References

Carroll, R. J., Ruppert, D., Crainiceanu, C. M., and Stefanski, L. A. (2006). Measurement error in nonlinear models: a modern perspective. Chapman and Hall/CRC.

Parast, L., Garcia, T. P., Prentice, R. L., & Carroll, R. J. (2022). Robust methods to correct for measurement error when evaluating a surrogate marker. Biometrics, 78(1), 9-23.

`R.multiple.surv` *Calculates the proportion of treatment effect explained by multiple surrogate markers measured at a specified time and primary outcome information up to that specified time*

Description

This function calculates the proportion of treatment effect on the primary outcome explained by multiple surrogate markers measured at t_0 and primary outcome information up to t_0. The user can also request a variance estimate, estimated using perturbating-resampling, and a 95% confidence interval. If a confidence interval is requested three versions are provided: a normal approximation based interval, a quantile based interval and Fieller's confidence interval, all using perturbation-resampling. The user can also request an estimate of the incremental value of the surrogate marker information.

Usage

```
R.multiple.surv(xone, xzero, deltaone, deltazero, sone, szero, type =1, t,
weight.perturb = NULL, landmark, extrapolate = FALSE, transform = FALSE,
conf.int = FALSE, var = FALSE, incremental.value = FALSE, approx = T)
```

Arguments

`xone`	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`xzero`	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`deltaone`	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`deltazero`	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
`sone`	matrix of numeric values; surrogate marker measurements at t_0 for treated observations. If $X_{1i} < t_0$, then the surrogate marker measurements should be NA.
`szero`	matrix of numeric values; surrogate marker measurements at t_0 for control observations. If $X_{0i} < t_0$, then the surrogate marker measurements should be NA.

type	type of estimate; options are 1 = two-stage robust estimator, 2 = weighted two-stage robust estimator, 3 = double-robust estimator, 4 = two-stage model-based estimator, 5 = weighted estimator, 6 = double-robust model-based estimator; default is 1.
t	the time of interest.
weight.perturb	weights used for perturbation resampling.
landmark	the landmark time t_0 or time of surrogate marker measurement.
extrapolate	TRUE or FALSE; indicates whether the user wants to use extrapolation.
transform	TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker pseudo-score.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval for delta is requested, default is FALSE.
var	TRUE or FALSE; indicates whether a variance estimate is requested, default is FALSE.
incremental.value	TRUE or FALSE; indicates whether the user would like to see the incremental value of the surrogate marker information, default is FALSE.
approx	TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

Details

Let $G \in \{A, B\}$ be the binary treatment indicator and we assume that subjects are randomly assigned to either treatment group A or B at baseline. Let T denote the time to the occurrence of the primary outcome, death for example, and $S = (S_1, S_2, ..., S_k)^T$ denote the vector of k surrogate markers measured at a given time t_0. Let $T^{(g)}$ and $S^{(g)}$ denote the counterfactual event time and surrogate marker measurements under treatment $G = g$ for $g \in \{A, B\}$. In practice, we only observe $(T, S) = (T^{(A)}, S^{(A)})$ or $(T^{(B)}, S^{(B)})$ depending on whether $G = A$ or B. The treatment effect, $\Delta(t)$, is the treatment difference in survival rates at time $t > t_0$, $\Delta(t) = E\{I(T^{(A)} > t)\} - E\{I(T^{(B)} > t)\} = P(T^{(A)} > t) - P(T^{(B)} > t)$ where $I(\cdot)$ is the indicator function. For individuals who are censored or experience the primary outcome before t_0, we assume that their S information is not available.

The surrogate marker information at time t_0 is defined as a combination of the observed information on $I(T > t_0)$ and the observed S at t_0, denoted by $Q_{t_0} = \{Q_{t_0}^{(g)}, g = A, B\}$, where $Q_{t_0}^{(g)} = \{T^{(g)} \wedge t_0, S^{(g)} I(T^{(g)} > t_0)\}$. With information on Q_{t_0}, the residual treatment effect is defined as: $\Delta_S(t, t_0) = E\{I(T^{(A)} > t) - I(T^{(B)} > t) \mid Q_{t_0}^{(A)} = Q_{t_0}^{(B)}\} = P(T^{(B)} > t_0) \int_S \psi_A(t \mid S, t_0) dF_B(S \mid t_0) - P(T^{(B)} > t)$ where $S = (s_1, ..., s_k)^T$, $F_g(S \mid t_0) = P(S^{(g)} \leq S \mid T^{(g)} > t_0)$, $\psi_g(t \mid S, t_0) = P(T^{(g)} > t \mid S^{(g)} = S, T^{(g)} > t_0)$. The proportion of the treatment effect on the primary outcome that is explained by the treatment effect on Q_{t_0} is $R_S(t, t_0) = 1 - \Delta_S(t, t_0)/\Delta$. This function provides 6 different estimators for $R_S(t, t_0)$ using censored data.

Due to censoring, the observed data consist of n observations $\{(G_i, X_i, \delta_i, S_i I(X_i > t_0)), i = 1, ..., n\}$ from the two treatment groups, where $X_i = \min(T_i, C_i)$, $\delta_i = I(T_i < C_i)$, and C_i denotes the censoring time for the ith subject. We assume independent censoring i.e., $(T_i, S_i) \perp$

$C_i \mid G_i$. For ease of notation, we also let $\{(X_{gi}, \delta_{gi}, S_{gi}I(X_{gi} > t_0)), i = 1, ..., n_g\}$ denote $n_g = \sum_{i=1}^{n} I(G_i = g)$ observations from treatment group $g \in \{A, B\}$, where $X_{gi} = \min(T_{gi}, C_{gi})$ and $\delta_{gi} = I(T_{gi} < C_{gi})$. Without loss of generality, we assume that $\bar{\pi}_g = n_g/n \to \pi_g \in (0, 1)$ as $n \to \infty$. Throughout, we estimate the treatment effect $\Delta(t) = P(T^{(A)} > t) - P(T^{(B)} > t)$ as $\hat{\Delta}(t) = n_A^{-1} \sum_{i=1}^{n_A} \frac{I(X_{Ai}>t)}{\hat{W}_A^C(t)} - n_B^{-1} \sum_{i=1}^{n_B} \frac{I(X_{Bi}>t)}{\hat{W}_B^C(t)}$ where $\hat{W}_g^C(t)$ is the Kaplan-Meier estimator of $W_g^C(t) = P(C_g > t)$.

We first describe the two-stage robust estimator which involves a two-stage procedure combining the use of a working model and a nonparametric estimation procedure for $\Delta_S(t, t_0)$. The idea is simply to summarize S into a univariate score U and then construct a nonparametric estimator for $R_S(t, t_0)$ treating U as S. To construct U, we approximate the conditional distribution of $T^{(A)} \mid S^{(A)}, T^{(A)} > t_0$ by using a working semiparametric model such as the landmark proportional hazards model $q_A(S) = P(T^{(A)} > t \mid T^{(A)} > t_0, S^{(A)}) = \exp\{-\Lambda_0(t|t_0)\exp(\beta^T S^{(A)})\}, t > t_0$, where $\Lambda_0(t|t_0)$ is the unspecified baseline cumulative hazard function for $T^{(A)}$ conditional on $\{T^{(A)} > t_0\}$ and β is an unknown vector of coefficients. Let $\hat{\beta}$ be the maximizer of the corresponding log partial likelihood function and $\hat{\Lambda}_0(t|t_0)$ be the Breslow-type estimate of baseline hazard. If one were to assume that this working model is correctly specified, then a consistent estimate of $\Delta_S(t, t_0)$ would simply be: $\hat{\Delta}_S^M = n_B^{-1} \sum_{i=1}^{n_B} [\exp\{-\hat{\Lambda}_0(t|t_0)\exp(\hat{\beta}^T S_{Bi})\} \frac{I(X_{Bi}>t_0)}{\hat{W}_B^C(t_0)} - \frac{I(X_{Bi}>t)}{\hat{W}_B^C(t)}]$. We refer to this estimate as the two-stage model-based estimator (option 4 for type). Instead of relying on correct specification of this model, we use the resulting score $U = \beta_0^T S$ as a univariate "pseudo-marker" to summarize the k surrogates. In the second stage, to estimate $\Delta_S(t, t_0)$, we apply a nonparametric approach with S represented by the univariate marker U. Specifically, we use a kernel Nelson-Aalen estimator to nonparametrically estimate $\phi_A(t|u, t_0) = P(T^{(A)} > t \mid U^{(A)} = u, T^{(A)} > t_0) = \exp\{-\Lambda_A(t|u, t_0)\}$ as $\hat{\phi}_A(t|u, t_0) = \exp\{-\hat{\Lambda}_A(t|u, t_0)\}$, where $\hat{\Lambda}_A(t|u, t_0) = \int_{t_0}^{t} \frac{\sum_{i=1}^{n_A} I(X_{Ai}>t_0) K_h\{\gamma(\hat{U}_{Ai}) - \gamma(u)\} dN_{Ai}(z)}{\sum_{i=1}^{n_A} I(X_{Ai}>t_0) K_h\{\gamma(\hat{U}_{Ai}) - \gamma(u)\} Y_{Ai}(z)}$, $\hat{U}_{Ai} = \hat{\beta}^T S_{Ai}$, $\hat{U}_{Bi} = \hat{\beta}^T S_{Bi}$, $Y_{Ai} = I(X_{Ai} \geq t)$, $N_{Ai}(t) = I(X_{Ai} \leq t)\delta_{Ai}$, $K(\cdot)$ is a smooth symmetric density function, $K_h(x) = K(x/h)/h$, and $\gamma(\cdot)$ is a given monotone transformation function. We then estimate $\Delta_S(t, t_0)$ as $\hat{\Delta}_S(t, t_0) = n_B^{-1} \sum_{i=1}^{n_B} [\hat{\phi}_A(t|\hat{U}_{Bi}, t_0) \frac{I(X_{Bi}>t_0)}{\hat{W}_B^C(t_0)} - \frac{I(X_{Bi}>t)}{\hat{W}_B^C(t)}]$ and $\hat{R}_S(t, t_0) = 1 - \hat{\Delta}_S(t, t_0)/\hat{\Delta}(t)$. We refer to this estimate as the two-stage robust estimator (option 1 for type).

The next estimator borrows ideas from the extensive causal inference literature focusing on double robust estimators two-stage weighted estimator with a propensity score weight explicitly balancing the two treatment groups with respect to the distribution of S. The weighting enables us to "adjust" the distribution of $S^{(A)}$ before constructing the conditional survival estimate $\hat{\phi}_A(t|u, t_0)$. This approach results in a double-robust estimator of $\Delta_S(t, t_0)$, which is consistent when either $U^{(A)}$ captures all the information about the relationship between $I(T^{(A)} \geq t)$ and $S^{(A)}$ or the propensity score model for $\pi(S, t_0) = P(G_i = B|S_i = S, T_i > t_0)$ is correctly specified. While $\pi(S, t_0)$ depends on t_0, for simplicity, we drop t_0 from our notation and simply use $\pi(S)$.

Regression models can be imposed to obtain estimates for $\pi(S)$. For example, a simple logistic regression model can be imposed for $\tilde{\pi}(S) = P(G_i = B|S_i = S, X_i > t_0)$ with $logit\{\tilde{\pi}(S)\} = \alpha_0 + \alpha_1^T S$, where α_0 and α_1 are estimated only among those with $X_{gi} > t_0$ to account for censoring. The propensity score of interest, $\pi(S)$, can be derived from $\tilde{\pi}(S)$ directly since $logit\{\pi(S)\} = logit\{\tilde{\pi}(S)\} + \log\{W_A^C(t_0)/W_B^C(t_0)\}$, which follows from the assumption that $(T_{gi}, S_{gi}) \perp C_{gi}$. We then modify the above expression by weighting observations with the estimated $L(S_{Ai}) =$

$\pi(S_{Ai})/\{1-\pi(S_{Ai})\}$ and obtain

$$\hat{\Lambda}_A^w(t|u,t_0) = \int_{t_0}^t \frac{\sum_{i=1}^{n_A} \hat{L}(S_{Ai}) I(X_{Ai}>t_0) K_h\{\gamma(\hat{U}_{Ai})-\gamma(u)\} dN_{Ai}(z)}{\sum_{i=1}^{n_A} \hat{L}(S_{Ai}) I(X_{Ai}>t_0) K_h\{\gamma(\hat{U}_{Ai})-\gamma(u)\} Y_{Ai}(z)},$$

, where $\hat{L}(S_{gi}) = \exp(\hat{\alpha}_0 + \hat{\alpha}_1^T S_{gi}) \hat{W}_B^C(t_0)/\hat{W}_A^C(t_0)$.

Subsequently, we define $\hat{\Delta}_S^w(t,t_0) = n_B^{-1} \sum_{i=1}^{n_B} [\hat{\phi}_A^w(t|\hat{U}_{Bi},t_0) \frac{I(X_{Bi}>t_0)}{\hat{W}_B^C(t_0)} - \frac{I(X_{Bi}>t)}{\hat{W}_B^C(t)}]$ and $\hat{R}_S^w(t,t_0) = 1 - \hat{\Delta}_S^w(t,t_0)/\hat{\Delta}(t)$ where $\hat{\phi}_A^w(t|t_0,u) = \exp\{-\hat{\Lambda}_A^w(t|t_0,u)\}$. We refer to this estimate as the weighted two-stage robust estimator (option 2 for type).

While the two-stage weighted estimator reflects one way to enhance the robustness of an initial estimator, the idea of combining a propensity-score type model and a regression-type model has certainly been extensively studied in the causal inference literature and a more familiar double-robust estimator can be constructed as: $\hat{\Delta}_S^{DR}(t,t_0) = n^{-1}[\sum_{i=1}^{n_A} \frac{I(X_{Ai}>t)}{\hat{W}_A^C(t)\bar{\pi}_B} \hat{L}(S_{Ai}) - \sum_{i=1}^{n_B} \frac{I(X_{Bi}>t)}{\hat{W}_B^C(t)\bar{\pi}_B}] - n^{-1}[\sum_{i=1}^{n_A} \frac{\hat{\phi}_A(t|\hat{U}_{Ai},t_0) I(X_{Ai}>t_0)}{\hat{W}_A^C(t_0)\bar{\pi}_B} \hat{L}(S_{Ai}) - \sum_{i=1}^{n_B} \frac{\hat{\phi}_A(t|\hat{U}_{Bi},t_0) I(X_{Bi}>t_0)}{\hat{W}_B^C(t_0)\bar{\pi}_B}]$ and $\hat{R}_S^{DR}(t,t_0) = 1 - \hat{\Delta}_S^{DR}(t,t_0)/\hat{\Delta}(t)$, where $\hat{\phi}_A(t|\hat{U}_{gi},t_0)$ is the (unweighted) estimate of $\phi_A(t \mid u,t_0)$ used in $\hat{\Delta}_S(t,t_0)$. We refer to this estimate as the double robust estimator (option 3 for type).

The weighted estimator (option type 5) is defined as: $\hat{\Delta}_S^{PS}(t,t_0) = n^{-1} \sum_{i=1}^{n} \{\frac{I(X_i>t)}{\hat{W}_{G_i}^C(t)\bar{\pi}_B}[I(G_i = A)\hat{L}(S_{Ai}) - I(G_i = B)]\}$ and $\hat{R}_S^{PS}(t,t_0) = 1 - \hat{\Delta}_S^{PS}(t,t_0)/\hat{\Delta}(t)$. This estimator completely relies on the correct specification of $\pi(S)$. The double-robust model-based estimator (option 6 for type) is defined as $\hat{\Delta}_S^{DR2}(t,t_0)$ and $\hat{R}_S^{DR2}(t,t_0) = 1 - \hat{\Delta}_S^{DR2}(t,t_0)/\hat{\Delta}(t)$ which are constructed parallel to the construction of $\hat{\Delta}_S^{DR}(t,t_0)$ i.e., a combination of $\hat{\Delta}_S^M(t,t_0)$ and $\hat{R}_S^{PS}(t,t_0)$.

Variance estimates are obtained using perturbation resampling. If a confidence interval is requested three versions are provided: a normal approximation based interval, a quantile based interval and Fieller's confidence interval, all using perturbation-resampling. An estimate of the incremental value of the surrogate marker information can also be requested; this essentially compared the proportion explained by the surrogate information vs. the proportion explained by T alone up to t_0. Details can be found in Parast, L., Cai, T., & Tian, L. (2021). Evaluating multiple surrogate markers with censored data. Biometrics, 77(4), 1315-1327.

Value

A list is returned:

`delta`	the estimate, $\hat{\Delta}(t)$, described in delta.estimate documentation.
`delta.s`	the residual treatment effect estimate, $\hat{\Delta}_S(t,t_0)$.
`R.s`	the estimated proportion of treatment effect explained by the set of markers, $\hat{R}_S(t,t_0)$.
`delta.var`	the variance estimate of $\hat{\Delta}(t)$; if var = TRUE or conf.int = TRUE.
`delta.s.var`	the variance estimate of $\hat{\Delta}_S(t,t_0)$; if var = TRUE or conf.int = TRUE.
`R.s.var`	the variance estimate of $\hat{R}_S(t,t_0)$; if var = TRUE or 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.delta`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}(t)$ based on sample quantiles of the perturbed values; if conf.int = TRUE.

`conf.int.normal.delta.s`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_S(t, t_0)$ based on a normal approximation; if conf.int = TRUE.

`conf.int.quantile.delta.s`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_S(t, t_0)$ based on sample quantiles of the perturbed values; if conf.int = TRUE.

`conf.int.normal.R.s`
: a vector of size 2; the 95% confidence interval for $\hat{R}_S(t, t_0)$ based on a normal approximation; if conf.int = TRUE.

`conf.int.quantile.R.s`
: a vector of size 2; the 95% confidence interval for $\hat{R}_S(t, t_0)$ based on sample quantiles of the perturbed values; if conf.int = TRUE.

`conf.int.fieller.R.s`
: a vector of size 2; the 95% confidence interval for $\hat{R}_S(t, t_0)$ based on Fieller's approach; if conf.int = TRUE.

`delta.t`
: the estimate, $\hat{\Delta}_T(t, t_0)$; if incremental.vaue = TRUE.

`R.t`
: the estimated proportion of treatment effect explained by survival only, $\hat{R}_T(t, t_0)$; if incremental.vaue = TRUE.

`incremental.value`
: the estimate of the incremental value of the surrogate markers, $\hat{IV}_S(t, t_0)$; if incremental.vaue = TRUE.

`delta.t.var`
: the variance estimate of $\hat{\Delta}_T(t, t_0)$; if var = TRUE or conf.int = TRUE and incremental.vaue = TRUE.

`R.t.var`
: the variance estimate of $\hat{R}_T(t, t_0)$; if var = TRUE or conf.int = TRUE and incremental.vaue = TRUE.

`incremental.value.var`
: the variance estimate of $\hat{IV}_S(t, t_0)$; if var = TRUE or conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.normal.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t, t_0)$ based on a normal approximation; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.quantile.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t, t_0)$ based on sample quantiles of the perturbed values; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.normal.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on a normal approximation; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.quantile.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on sample quantiles of the perturbed values; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.fieller.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on Fieller's approach; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.normal.iv`
: a vector of size 2; the 95% confidence interval for $I\hat{V}_S(t, t_0)$ based on a normal approximation; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.quantile.iv`
: a vector of size 2; the 95% confidence interval for $I\hat{V}_S(t, t_0)$ based on sample quantiles of the perturbed values; if conf.int = TRUE and incremental.vaue = TRUE.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting".

Author(s)

Layla Parast

References

Parast, L., Cai, T., & Tian, L. (2017). Evaluating surrogate marker information using censored data. Statistics in Medicine, 36(11), 1767-1782.

Parast, L., Cai, T., & Tian, L. (2021). Evaluating multiple surrogate markers with censored data. Biometrics, 77(4), 1315-1327.

Examples

```
data(d_example_multiple)
names(d_example_multiple)
## Not run:
R.multiple.surv(xone = d_example_multiple$x1, xzero = d_example_multiple$x0, deltaone =
d_example_multiple$delta1, deltazero = d_example_multiple$delta0, sone =
as.matrix(d_example_multiple$s1), szero = as.matrix(d_example_multiple$s0),
type =1, t = 1, landmark=0.5)
R.multiple.surv(xone = d_example_multiple$x1, xzero = d_example_multiple$x0, deltaone =
d_example_multiple$delta1, deltazero = d_example_multiple$delta0, sone =
as.matrix(d_example_multiple$s1), szero = as.matrix(d_example_multiple$s0),
type =1, t = 1, landmark=0.5, conf.int = T)
R.multiple.surv(xone = d_example_multiple$x1, xzero = d_example_multiple$x0, deltaone =
d_example_multiple$delta1, deltazero = d_example_multiple$delta0, sone =
as.matrix(d_example_multiple$s1), szero = as.matrix(d_example_multiple$s0),
type =3, t = 1, landmark=0.5)

## End(Not run)
```

R.s.estimate *Calculates the proportion of treatment effect explained*

Description

This function calculates the proportion of treatment effect on the primary outcome explained by the treatment effect on the surrogate marker(s). This function is intended to be used for a fully observed continuous outcome. The user can also request a variance estimate and a 95% confidence interval, both estimated using perturbating-resampling. If a confidence interval is requested three versions are provided: a normal approximation based interval, a quantile based interval, and Fieller's confidence interval.

Usage

```
R.s.estimate(sone, szero, yone, yzero, var = FALSE, conf.int = FALSE,
weight.perturb = NULL, number = "single", type = "robust",extrapolate = FALSE,
transform = FALSE,warn.te = FALSE, warn.support = FALSE)
```

Arguments

sone	numeric vector or matrix; surrogate marker for treated observations, assumed to be continuous. If there are multiple surrogates then this should be a matrix with n_1 (number of treated observations) rows and n.s (number of surrogate markers) columns.
szero	numeric vector; surrogate marker for control observations, assumed to be continuous.If there are multiple surrogates then this should be a matrix with n_0 (number of control observations) rows and n.s (number of surrogate markers) columns.
yone	numeric vector; primary outcome for treated observations, assumed to be continuous.
yzero	numeric vector; primary outcome for control observations, assumed to be continuous.
var	TRUE or FALSE; indicates whether a variance estimate is requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval is requested, default is FALSE
weight.perturb	a $n_1 + n_0$ by x matrix of weights where $n_1 =$ length of yone and $n_0 =$ length of yzero; used for perturbation-resampling, default is null.
number	specifies the number of surrogate markers; choices are "multiple" or "single", default is "single"
type	specifies the type of estimation; choices are "robust" or "model" or "freedman", default is "robust"
extrapolate	TRUE or FALSE; indicates whether the user wants to use extrapolation.

`transform`	TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker.
`warn.te`	value to control warnings; user does not need to specify.
`warn.support`	value to control warnings; user does not need to specify.

Details

Let $Y^{(1)}$ and $Y^{(0)}$ denote the primary outcome under the treatment and primary outcome under the control,respectively. Let $S^{(1)}$ and $S^{(0)}$ denote the surrogate marker under the treatment and the surrogate marker under the control,respectively. The residual treatment effect is defined as

$$\Delta_S = \int_{-\infty}^{\infty} E(Y^{(1)}|S^{(1)} = s)dF_0(s) - \int_{-\infty}^{\infty} E(Y^{(0)}|S^{(0)} = s)dF_0(s),$$

where $\Delta_S(s) = E(Y^{(1)}|S^{(1)} = s) - E(Y^{(0)}|S^{(0)} = s)$ and $F_0(\cdot)$ is the marginal cumulative distribution function of $S^{(0)}$, the surrogate marker measure under the control. The proportion of treatment effect explained by the surrogate marker, which we denote by R_S, can be expressed using a contrast between Δ_S and Δ:

$$R_S = \{\Delta - \Delta_S\}/\Delta = 1 - \Delta_S/\Delta.$$

The definition and estimation of Δ is described in the delta.estimate documentation.

A flexible model-based approach to estimate Δ_S in the single marker setting is to specify:

$$E(S^{(0)}) = \alpha_0 \quad \text{and} \quad E(S^{(1)}) - E(S^{(0)}) = \alpha_1,$$

$$E(Y^{(0)}|S^{(0)}) = \beta_0 + \beta_1 S^{(0)} \quad \text{and} \quad E(Y^{(1)}|S^{(1)}) = (\beta_0 + \beta_2) + (\beta_1 + \beta_3)S^{(1)}.$$

It can be shown that when these models hold, $\Delta_S = \beta_2 + \beta_3\alpha_0$. Thus, reasonable estimates for Δ_S and R_S using this approach would be $\hat{\Delta}_S = \hat{\beta}_2 + \hat{\beta}_3\hat{\alpha}_0$ and $\hat{R}_S = 1 - \hat{\Delta}_S/\hat{\Delta}$.

For robust estimation of Δ_S in the single marker setting, we estimate $\mu_1(s) = E(Y^{(1)}|S^{(1)} = s)$ nonparametrically using kernel smoothing:

$$\hat{\mu}_1(s) = \frac{\sum_{i=1}^{n_1} K_h\left(S_{1i} - s\right) Y_{1i}}{\sum_{i=1}^{n_1} K_h\left(S_{1i} - s\right)}$$

where S_{1i} is the observed $S^{(1)}$ for person i, Y_{1i} is the observed $Y^{(1)}$ for person i, $K(\cdot)$ is a smooth symmetric density function with finite support, $K_h(\cdot) = K(\cdot/h)/h$ and h is a specified bandwidth. As in most nonparametric functional estimation procedures, the choice of the smoothing parameter h is critical. To eliminate the impact of the bias of the conditional mean function on the resulting estimator, we require the standard undersmoothing assumption of $h = O(n_1^{-\delta})$ with $\delta \in (1/4, 1/3)$. To obtain an appropriate h we first use bw.nrd to obtain h_{opt}; and then we let $h = h_{opt}n_1^{-c_0}$ with $c_0 = 0.25$. We then estimate Δ_S as

$$\hat{\Delta}_S = \sum_{i=1}^{n_0} \frac{\hat{\mu}_1(S_{0i}) - Y_{0i}}{n_0}$$

where S_{0i} is the observed $S^{(0)}$ for person i and Y_{0i} is the observed $Y^{(0)}$ for person i. Lastly, we estimate R_S as $\hat{R}_S = 1 - \hat{\Delta}_S/\hat{\Delta}$.

This function also allows for estimation of R_S using Freedman's approach. Let Y denote the primary outcome, S denote the surrogate marker, and G denote the treatment group (0 for control, 1 for treatment). Freedman's approach to calculating the proportion of treatment effect explained by the surrogate marker is to fit the following two regression models:

$$E(Y|G) = \gamma_0 + \gamma_1 I(G=1) \quad \text{and} \quad E(Y|G,S) = \gamma_{0S} + \gamma_{1S} I(G=1) + \gamma_{2S} S$$

and estimating the proportion of treatment effect explained, denoted by R_S, as $1 - \hat{\gamma}_{1S}/\hat{\gamma}_1$.

This function also estimates R_S in a multiple marking setting. A flexible model-based approach to estimate Δ_S in the multiple marker setting is to specify models for $E(Y|G,S)$ and $E(S_j|G)$ for each S_j in $S = \{S_1, ...S_p\}$ (where p is the number of surrogate markers). Without loss of generality, consider the case where there are three surrogate markers, $S = \{S_1, S_2, S_3\}$ and one specifies the following linear models:

$$E(Y^{(0)}|S^{(0)}) = \beta_0 + \beta_1 S_1^{(0)} + \beta_2 S_2^{(0)} + \beta_3 S_3^{(0)}$$

$$E(Y^{(1)}|S^{(1)}) = (\beta_0 + \beta_4) + (\beta_1 + \beta_5) S_1^{(1)} + (\beta_2 + \beta_6) S_2^{(1)} + (\beta_3 + \beta_7) S_3^{(1)}$$

$$E(S_j^{(0)}) = \alpha_j, \quad j = 1, 2, 3.$$

It can be shown that when these models hold

$$\Delta_S = \beta_4 + \beta_5 \alpha_1 + \beta_6 \alpha_2 + \beta_7 \alpha_3.$$

Thus, reasonable estimates for Δ_S and R_S here would be easily obtained by replacing the unknown regression coefficients in the models above by their consistent estimators.

For robust estimation of S Δ_S in the multiple marker setting, we use a two-stage procedure combining the model-based approach and the nonparametric estimation procedure from the single marker setting. Specifically, we use a working semiparametric model:

$$E(Y^{(1)}|S^{(1)} = S) = \beta_0 + \beta_1 S_1^{(1)} + \beta_2 S_2^{(1)} + \beta_3 S_3^{(1)}$$

and define $Q^{(1)} = \hat{\beta}_0 + \hat{\beta}_1 S_1^{(1)} + \hat{\beta}_2 S_2^{(1)} + \hat{\beta}_3 S_3^{(1)}$ and $Q^{(0)} = \hat{\beta}_0 + \hat{\beta}_1 S_1^{(0)} + \hat{\beta}_2 S_2^{(0)} + \hat{\beta}_3 S_3^{(0)}$ to reduce the dimension of S in the first stage and in the second stage, we apply the robust approach used in the single marker setting to estimate its surrogacy.

To use Freedman's approach in the presence of multiple markers, the markers are simply additively entered into the second regression model.

Variance estimation and confidence interval construction are performed using perturbation-resampling. Specifically, let $\left\{ V^{(b)} = (V_{11}^{(b)}, ...V_{1n_1}^{(b)}, V_{01}^{(b)}, ...V_{0n_0}^{(b)})^T, b = 1, ..., D \right\}$ be $n \times D$ independent copies of a positive random variables V from a known distribution with unit mean and unit variance. Let

$$\hat{\Delta}^{(b)} = \frac{\sum_{i=1}^{n_1} V_{1i}^{(b)} Y_{1i}}{\sum_{i=1}^{n_1} V_{1i}^{(b)}} - \frac{\sum_{i=1}^{n_0} V_{0i}^{(b)} Y_{0i}}{\sum_{i=1}^{n_0} V_{0i}^{(b)}}.$$

The variance of $\hat{\Delta}$ is obtained as the empirical variance of $\{\hat{\Delta}^{(b)}, b = 1, ..., D\}$. In this package, we use weights generated from an Exponential(1) distribution and use $D = 500$. Variance estimates for $\hat{\Delta}_S$ and $\hat{R}_S$ are calculated similarly. We construct two versions of the 95% confidence interval for each estimate: one based on a normal approximation confidence interval using the estimated

variance and another taking the 2.5th and 97.5th empirical percentile of the perturbed quantities. In addition, we use Fieller's method to obtain a third confidence interval for R_S as

$$\left\{1 - r : \frac{(\hat{\Delta}_S - r\hat{\Delta})^2}{\hat{\sigma}_{11} - 2r\hat{\sigma}_{12} + r^2\hat{\sigma}_{22}} \leq c_\alpha\right\},$$

where $\hat{\Sigma} = (\hat{\sigma}_{ij})_{1\leq i,j\leq 2}$ and c_α is the $(1-\alpha)$th percentile of

$$\left\{\frac{\{\hat{\Delta}_S^{(b)} - (1-\hat{R}_S)\hat{\Delta}^{(b)}\}^2}{\hat{\sigma}_{11} - 2(1-\hat{R}_S)\hat{\sigma}_{12} + (1-\hat{R}_S)^2\hat{\sigma}_{22}}, b = 1, \cdots, C\right\}$$

where $\alpha = 0.05$.

Note that if the observed supports for S are not the same, then $\hat{\mu}_1(s)$ for $S_{0i} = s$ outside the support of S_{1i} may return NA (depending on the bandwidth). If extrapolation = TRUE, then the $\hat{\mu}_1(s)$ values for these surrogate values are set to the closest non-NA value. If transform = TRUE, then S_{1i} and S_{0i} are transformed such that the new transformed values, S_{1i}^{tr} and S_{0i}^{tr} are defined as: $S_{gi}^{tr} = F([S_{gi} - \mu]/\sigma)$ for $g = 0, 1$ where $F(\cdot)$ is the cumulative distribution function for a standard normal random variable, and μ and σ are the sample mean and standard deviation, respectively, of $(S_{1i}, S_{0i})^T$.

Value

A list is returned:

`R.s` the estimate, $\hat{R}_S$, described above.

`R.s.var` the variance estimate of $\hat{R}_S$; if var = TRUE or conf.int = TRUE.

`conf.int.normal.R.s` a vector of size 2; the 95% confidence interval for $\hat{R}_S$ based on a normal approximation; if conf.int = TRUE.

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

`conf.int.fieller.R.s` a vector of size 2; the 95% confidence interval for $\hat{R}_S$ based on Fieller's approach, described above; if conf.int = TRUE.

For all options other then "freedman", the following are also returned:

`delta` the estimate, $\hat{\Delta}$, described in delta.estimate documentation.

`delta.s` the estimate, $\hat{\Delta}_S$, described above.

`delta.var` the variance estimate of $\hat{\Delta}$; if var = TRUE or conf.int = TRUE.

`delta.s.var` the variance estimate of $\hat{\Delta}_S$; if var = TRUE or conf.int = TRUE.

`conf.int.normal.delta` a vector of size 2; the 95% confidence interval for $\hat{\Delta}$ based on a normal approximation; if conf.int = TRUE.

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

`conf.int.normal.delta.s`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_S$ based on a normal approximation; if conf.int = TRUE.

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

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the proportion of treatment effect explained in this setting". In the single marker case with the robust estimation approach, if the observed support of the surrogate marker for the control group is outside the observed support of the surrogate marker for the treatment group, the user will receive the following message: "Warning: observed supports do not appear equal, may need to consider a transformation or extrapolation"

Author(s)

Layla Parast

References

Freedman, L. S., Graubard, B. I., & Schatzkin, A. (1992). Statistical validation of intermediate endpoints for chronic diseases. Statistics in medicine, 11(2), 167-178.

Parast, L., McDermott, M., Tian, L. (2016). Robust estimation of the proportion of treatment effect explained by surrogate marker information. Statistics in Medicine, 35(10):1637-1653.

Wang, Y., & Taylor, J. M. (2002). A measure of the proportion of treatment effect explained by a surrogate marker. Biometrics, 58(4), 803-812.

Fieller, Edgar C. (1954). Some problems in interval estimation. Journal of the Royal Statistical Society. Series B (Methodological), 175-185.

Fieller, E. C. (1940). The biological standardization of insulin. Supplement to the Journal of the Royal Statistical Society, 1-64.

Examples

```
data(d_example)
names(d_example)
R.s.estimate(yone=d_example$y1, yzero=d_example$y0, sone=d_example$s1.a, szero=d_example$s0.a,
number = "single", type = "robust")
R.s.estimate(yone=d_example$y1, yzero=d_example$y0, sone=cbind(d_example$s1.a,d_example$s1.b,
d_example$s1.c), szero=cbind(d_example$s0.a, d_example$s0.b, d_example$s0.c),
number = "multiple", type = "model")
```

R.s.estimate.me *Calculates the proportion of treatment effect explained correcting for measurement error in the surrogate marker*

Description

This function calculates the proportion of treatment effect on the primary outcome explained by the treatment effect on a surrogate marker, correcting for measurement error in the surrogate marker. This function is intended to be used for a fully observed continuous outcome. The user must specify what type of estimation they would like (parametric or nonparametric estimation of the proportion explained, denoted by R) and what estimator they would like (see below for details).

Usage

```
R.s.estimate.me(sone, szero, yone, yzero, parametric = FALSE, estimator = "n",
me.variance, extrapolate = TRUE, transform = FALSE, naive = FALSE, Ronly = TRUE)
```

Arguments

sone	numeric vector or matrix; surrogate marker for treated observations, assumed to be continuous. If there are multiple surrogates then this should be a matrix with n_1 (number of treated observations) rows and n.s (number of surrogate markers) columns.
szero	numeric vector; surrogate marker for control observations, assumed to be continuous.If there are multiple surrogates then this should be a matrix with n_0 (number of control observations) rows and n.s (number of surrogate markers) columns.
yone	numeric vector; primary outcome for treated observations, assumed to be continuous.
yzero	numeric vector; primary outcome for control observations, assumed to be continuous.
parametric	TRUE or FALSE; indicates whether the user wants the parametric approach to be used (TRUE) or nonparametric (FALSE).
estimator	options are "d","q","n" for parametric and "q","n" for nonparametric; "d" stands for the disattenuated estimator, "q" stands for the SIMEX estimator with quadratic extrapolation, "n" stands for the SIMEX estimator with a nonlinear extrapolation. Note that the nonlinear extrapolation may have convergence issues with a small sample size; if this occurs, please consider using quadratic extrapolation instead.
me.variance	the variance of the measurement error; must be provided.
extrapolate	TRUE or FALSE; indicates whether the user wants to use extrapolation.
transform	TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker.
naive	TRUE or FALSE; indicates whether the user wants the naive estimate (not correcting for measurement error) to also be calculated

`Ronly` TRUE or FALSE; indicates whether the user wants only R (and corresponding variance and confidence intervals) to be returned.

Details

While there are many methods available to quantify the value of a surrogate marker, most assume that the marker is measured without error. This function calculates the proportion of treatment effect on the primary outcome explained by the treatment effect on a surrogate marker, correcting for measurement error in the surrogate marker. The user can choose either the parametric framework or nonparametric framework for estmation. Within the parametric framework there are three options for measurement error correction: the disattenuated estimator, the SIMEX estimator with quadratic extrapolation, and the SIMEX estimator with nonlinear extrapolation. Within the nonparametric framework there are two options for measurement error correction: the SIMEX estimator with quadratic extrapolation and the SIMEX estimator with nonlinear extrapolation. We describe each below.

Let G be the binary treatment indicator with $G = 1$ indicating treatment and $G = 0$ indicating control (or placebo). We assume throughout that subjects are randomly assigned to treatment or control at baseline. Let Y and S denote the continuous primary outcome and continuous surrogate marker, respectively, where S is measured post-baseline and is assumed to be a biomarker, clinical measurement, psychological test score, or other physiological measurement. In the absence of measurement error, the observed data consists of $\{Y_i, S_i, G_i\}$ for $i \in \{1, ..., n\}$. With measurement error, instead of observing S we observe $W = S + U$, where $E(U|S) = 0$ and the variance of U is σ_u^2. Such measurement error may be attributable to, for example, laboratory error. Thus, our observed data will consist of $\{Y_i, W_i, G_i\}$ for $i \in \{1, ..., n\}$. Throughout, we assume that σ_u^2 is known. Here, we are interested in estimating the proprtion of the treatment effect on the primary outcome that is explained by the treatment effect on the surrogate marker, denoted as R_S.

To estimate R_S parametrically, we assume the following models $E(Y|G) = \beta_0 + \beta_1 G$ and $E(Y|G, S) = \beta_0^* + \beta_1^* G + \beta_2^* S$. It can be shown that if these models hold, $R_S = 1 - \beta_1^*/\beta_1$. When $W = S + U$ is available instead of S, this measurement error does not affect estimation of β_1, but it does affect estimation of β_1^*, and β_2^*. Since estimation of R_S relies on estimation of β_1 and β_1^*, we focus on the effect of measurement error on β_1^* estimation. The attenuation bias for $\hat{\beta}_1^*$ and $\hat{R}$ can be written out in closed form when the proportion of treatment effect is parametrically estimated as described above, when these specified models hold, and when the surrogate marker S is measured with error. There exist two methods to eliminate this bias when estimating R_S. Taking advantage of the fact that we can express the attenuation bias in closed form, the first is a straightforward disattenuated estimator: $\hat{\beta}_{1A} = \hat{\beta}_1^* - \frac{\hat{\beta}_2^* \{\Omega_W^2 \Omega_{GW} - \Omega_{GW}(\Omega_W^2 - \sigma_u^2)\}}{\Omega_G^2(\Omega_W^2 - \sigma_u^2) - \Omega_{GW}\Omega_{GW}}$ and $\hat{R}_A = 1 - \left[\hat{\beta}_1^* - \frac{\hat{\beta}_2^* \{\Omega_W^2 \Omega_{GW} - \Omega_{GW}(\Omega_W^2 - \sigma_u^2)\}}{\Omega_G^2(\Omega_W^2 - \sigma_u^2) - \Omega_{GW}\Omega_{GW}}\right] / \hat{\beta}_1$ where Ω^2 denotes the sample variance or covariance.

The second method to eliminate this bias uses Simulation Extrapolation (SIMEX) estimation, which is a simulation-based method that involves first generating additional measurement error and observing how it affects the bias of the parameter estimate of interest, and then extrapolating this information to a setting with no measurement error. To incorporate SIMEX estimation within our surrogate marker framework, we define $W_{b,i}(\lambda) = W_i + \lambda^{1/2} \sigma_u \epsilon_{i,b}$ for $b = 1, ..., B$ where $B = 50$, $\epsilon_{i,b} \sim N(0, 1)$, σ_u is assumed known, and $\lambda \in (0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0)$ and for each iteration b and λ value, obtaining $\hat{\beta}_{1b}^*(\lambda)$ by fitting the regression model: $E(Y \mid W_b(\lambda), S) = \beta_{0b}^* + \beta_{1b}^* W_b(\lambda) + \beta_{2b}^* S$. We then calculate the average estimate for each quantity over the iterations $b = 1, ..., B$ for each λ value, denoted as $\hat{\beta}_{1,S,\sigma_u^2(1+\lambda)}^* = \sum_{b=1}^B \hat{\beta}_{1b}^*(\lambda)$. The second step, extrapolation, takes these average estimates for each λ value and extrapolates using a

function $G(\Gamma, \lambda)$ to obtain the estimated quantity if $\lambda = -1$. For the extrapolation step, we use both a quadratic extrapolation and nonlinear extrapolation i.e., we solve for $\Gamma = (\alpha_0, \alpha_1, \alpha_2)^T$ in $\hat{\beta}^*_{1,S,\sigma^2_u(1+\lambda)} = \alpha_0 + \alpha_1\lambda + \alpha_2\lambda^2$ and $\hat{\beta}^*_{1,S,\sigma^2_u(1+\lambda)} = \alpha_0 + \alpha_1/(\alpha_2 + \lambda)$, respectively. Using the estimates of $\alpha_0, \alpha_1, \alpha_2$, we calculate the predicted $\hat{\beta}^*_{1,S,\sigma^2_u(1+\lambda)}$ when $\lambda = -1$. In essence, the simulations add successively larger measurement errors of size $(1 + \lambda)\sigma^2_u$ and then extrapolate to the case when $\lambda = -1$ such that the measurement error is 0. We denote the resulting estimator of β^*_1 as $\hat{\beta}^*_{1,SIMEX} = G(\hat{\Gamma}, -1)$ and define $\hat{R}_{SIMEX} = 1 - \hat{\beta}^*_{1,SIMEX}/\hat{\beta}_1$.

While the parametric approach to estimate the proportion of treatment effect explained by S is most commonly used in clinical practice, previous work has demonstrated biased results when the assumed models are not correctly specified. An alternative approach involves estimating the treatment effect, Δ, and residual treatment effect, Δ_S, as R_S is defined as $1 - \Delta/\Delta_S$. The quantity Δ can be estimated simply by $\hat{\Delta} = n_1^{-1}\sum_{i=1}^n Y_i I(G_i = 1) - n_0^{-1}\sum_{i=1}^n Y_i I(G_i = 0)$, where n_1 and n_0 denote the number of individuals in the treatment and control groups, respectively. The quantity Δ_S can be estimated nonparametrically using kernel smoothing as $\hat{\Delta}_S = n_0^{-1}\sum_{i:G_i=0}\hat{\mu}_1(S_i) - n_0^{-1}\sum_{i=1}^n Y_i I(G_i = 0)$ where $\hat{\mu}_1(s) = \{\sum_{j:G_j=1} K_h(S_j - s)Y_j\}/\{\sum_{j:G_j=1} K_h(S_j - s)\}$, $K(\cdot)$ is a smooth symmetric density function with finite support, $K_h(\cdot) = K(\cdot/h)/h$ and h is a specified bandwidth such that $h = O(n_1^{-\nu})$ with $\nu \in (1/4, 1/2)$.

When $W = S + U$ is available instead of S, estimation of Δ is not affected whereas estimation of Δ_S is affected and thus, the nonparametric estimation procedure described above results in a biased estimate of R_S. Unlike the parametric approach, the attenuation bias cannot be expressed in closed form. Within this nonparametric framework, SIMEX estimation can be used to correct for measurement error. We implement the estimation procedure as described above where we first generate additional measurement error to obtain $W_{b,i}(\lambda)$ and for each iteration b and λ values obtain $\hat{\Delta}_{S,b}(\lambda) = n_0^{-1}\sum_{i:G_i=0}\left\{\frac{\sum_{j:G_j=1} K_h(W_{b,j}(\lambda) - W_{b,i}(\lambda))Y_j}{\sum_{j:G_j=1} K_h(W_{b,j}(\lambda) - W_{b,i}(\lambda))}\right\} - n_0^{-1}\sum_{i=1}^n Y_i I(G_i = 0)$. We then calculate the average estimate for each quantity over the iterations $b = 1, ..., B$ for each λ value, denoted as $\hat{\Delta}_{S,\sigma^2_u(1+\lambda)} = \sum_{b=1}^B \hat{\Delta}_{S,b}(\lambda)$ and extrapolate using a function $G(\Gamma, \lambda)$; we specifically use the quadratic and nonlinear functions as in the parametric setting. We denote the resulting estimator of Δ_S as $\hat{\Delta}_{S,SIMEX} = G(\hat{\Gamma}, -1)$ and define $\hat{R}_{S,SIMEX} = 1 - \hat{\Delta}_{S,SIMEX}/\hat{\Delta}$.

In this function, parametric estimation is equivalent to Freedman's approach in the R.s.estimate documentation; nonparametric estimation is equivalent to the robust approach in the R.s.estimate documentation. Variance estimates for all estimators are calculated in this function based on derived closed form variance expressions. For all approaches, confidence intervals for Δ_S can be constructed using a normal approximation; confidence intervals for R_S can be constructed using either a normal approximation or using Fieller's method, all of which are provided in this function. Details regarding the asymptotic properties of these estimators and closed form variance calculation can be found in: Parast, L., Garcia, T. P., Prentice, R. L., & Carroll, R. J. (2022). Robust methods to correct for measurement error when evaluating a surrogate marker. Biometrics, 78(1), 9-23.

Value

A list is returned:

`R.naive`	the naive estimate of the proportion of treatment effect explained by the surrogate marker; only if naive = TRUE
`R.naive.var`	the estimated variance of the naive estimate of the proportion of treatment effect explained by the surrogate marker; only if naive = TRUE

`R.naive.CI.normal`
: the 95% confidence interval using the normal approximation for the naive estimate of the proportion of treatment effect explained by the surrogate marker; only if naive = TRUE

`R.naive.CI.fieller`
: the 95% confidence interval using Fieller's approach for the naive estimate of the proportion of treatment effect explained by the surrogate marker; only if naive = TRUE

`B1star.naive`
: the naive estimate of the adjusted regression coefficient for treatment; only if naive = TRUE and Ronly = FALSE and parametric = TRUE

`B1star.naive.var`
: the estimated variance of the naive estimate of the adjusted regression coefficient for treatment; only if naive = TRUE and Ronly = FALSE and parametric = TRUE

`B1star.naive.CI.normal`
: the 95% confidence interval using the normal approximation for the naive estimate of the adjusted regression coefficient for treatment; only if naive = TRUE and Ronly = FALSE and parametric = TRUE

`deltas.naive`
: the naive estimate of the residual treatment effect; only if naive = TRUE and Ronly = FALSE and parametric = FALSE

`deltas.naive.var`
: the estimated variance of the naive estimate of the residual treatment effect; only if naive = TRUE and Ronly = FALSE and parametric = FALSE

`deltas.naive.CI.normal`
: the 95% confidence interval using the normal approximation for the naive estimate of the residual treatment effect; only if naive = TRUE and Ronly = FALSE and parametric = FALSE

`R.corrected.dis`
: the corrected disattenuated estimate of the proportion of treatment effect explained by the surrogate marker; only if parametric = TRUE and estimator ="d"

`R.corrected.var.dis`
: the estimated variance of the corrected disattenuated estimate of the proportion of treatment effect explained by the surrogate marker; only if naive = TRUE

`R.corrected.CI.normal.dis`
: the 95% confidence interval using the normal approximation for the corrected disattenuated estimate of the proportion of treatment effect explained by the surrogate marker; only if parametric = TRUE and estimator ="d"

`R.corrected.CI.fieller.dis`
: the 95% confidence interval using Fieller's approach for the corrected disattenuated estimate of the proportion of treatment effect explained by the surrogate marker; only if parametric = TRUE and estimator ="d"

`B1star.corrected.dis`
: the corrected disattenuated estimate of the adjusted regression coefficient for treatment; only if parametric = TRUE and estimator = "d" and Ronly = FALSE

`B1star.corrected.var.dis`
: the estimated variance of the corrected disattenuated estimate of the adjusted regression coefficient for treatment; only if parametric = TRUE and estimator = "d" and Ronly = FALSE

`B1star.corrected.CI.normal.dis`
: the 95% confidence interval using the normal approximation for the corrected disattenuated estimate of the adjusted regression coefficient for treatment; only if parametric = TRUE and estimator = "d" and Ronly = FALSE

`R.corrected.q`
: the corrected SIMEX (quadratic) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`R.corrected.var.q`
: the estimated variance of the corrected SIMEX (quadratic) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`R.corrected.CI.normal.q`
: the 95% confidence interval using the normal approximation for the corrected SIMEX (quadratic) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`R.corrected.CI.fieller.q`
: the 95% confidence interval using Fieller's approach for the corrected SIMEX (quadratic) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`B1star.corrected.q`
: the corrected SIMEX (quadratic) estimate of the adjusted regression coefficient for treatment; only if estimator = "q" and Ronly = FALSE and parametric = TRUE

`B1star.corrected.var.q`
: the estimated variance of the corrected SIMEX (quadratic) estimate of the adjusted regression coefficient for treatment; only if estimator = "q" and Ronly = FALSE and parametric = TRUE

`B1star.corrected.CI.normal.q`
: the 95% confidence interval using the normal approximation for the corrected SIMEX (quadratic) estimate of the adjusted regression coefficient for treatment; only if estimator = "q" and Ronly = FALSE and parametric = TRUE

`deltas.corrected.q`
: the corrected SIMEX (quadratic) estimate of the residual treatment effect; only if estimator = "q" and Ronly = FALSE and parametric = FALSE

`deltas.corrected.var.q`
: the estimated variance of the corrected SIMEX (quadratic) estimate of the residual treatment effect; only if estimator = "q" and Ronly = FALSE and parametric = FALSE

`deltas.corrected.CI.normal.q`
: the 95% confidence interval using the normal approximation for the corrected SIMEX (quadratic) estimate of the residual treatment effect; only if estimator = "q" and Ronly = FALSE and parametric = FALSE

`R.corrected.nl`
: the corrected SIMEX (nonlinear) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`R.corrected.var.nl`
: the estimated variance of the corrected SIMEX (nonlinear) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`R.corrected.CI.normal.nl`
: the 95% confidence interval using the normal approximation for the corrected SIMEX (nonlinear) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`R.corrected.CI.fieller.nl`
: the 95% confidence interval using Fieller's approach for the corrected SIMEX (nonlinear) estimate of the proportion of treatment effect explained by the surrogate marker; only if estimator = "q"

`B1star.corrected.nl`
: the corrected SIMEX (nonlinear) estimate of the adjusted regression coefficient for treatment; only if estimator = "q" and Ronly = FALSE and parametric = TRUE

`B1star.corrected.var.nl`
: the estimated variance of the corrected SIMEX (nonlinear) estimate of the adjusted regression coefficient for treatment; only if estimator = "q" and Ronly = FALSE and parametric = TRUE

`B1star.corrected.CI.normal.nl`
: the 95% confidence interval using the normal approximation for the corrected SIMEX (nonlinear) estimate of the adjusted regression coefficient for treatment; only if estimator = "q" and Ronly = FALSE and parametric = TRUE

`deltas.corrected.nl`
: the corrected SIMEX (nonlinear) estimate of the residual treatment effect; only if estimator = "q" and Ronly = FALSE and parametric = FALSE

`deltas.corrected.var.nl`
: the estimated variance of the corrected SIMEX (nonlinear) estimate of the residual treatment effect; only if estimator = "q" and Ronly = FALSE and parametric = FALSE

`deltas.corrected.CI.normal.nl`
: the 95% confidence interval using the normal approximation for the corrected SIMEX (nonlinear) estimate of the residual treatment effect; only if estimator = "q" and Ronly = FALSE and parametric = FALSE

Author(s)

Layla Parast

References

Parast, L., Garcia, T. P., Prentice, R. L., & Carroll, R. J. (2022). Robust methods to correct for measurement error when evaluating a surrogate marker. Biometrics, 78(1), 9-23.

Examples

```
data(d_example_me)
names(d_example_me)
R.s.estimate.me(yone=d_example_me$y1, yzero=d_example_me$y0, sone=d_example_me$s1,
szero=d_example_me$s0, parametric = TRUE, estimator = "d", me.variance = 0.5,
naive= TRUE, Ronly = FALSE)
R.s.estimate.me(yone=d_example_me$y1, yzero=d_example_me$y0, sone=d_example_me$s1,
```

```
szero=d_example_me$s0, parametric = TRUE, estimator = "q", me.variance = 0.5,
naive= FALSE, Ronly = TRUE)

#estimating measurement error variance with replicates
replicates = rbind(cbind(d_example_me$s1_rep1, d_example_me$s1_rep2,
d_example_me$s1_rep3), cbind(d_example_me$s0_rep1, d_example_me$s0_rep2,
d_example_me$s0_rep3))
mean.i = apply(replicates,1,mean, na.rm = TRUE)
num.i = apply(replicates,1,function(x) sum(!is.na(x)))
var.u = sum((replicates-mean.i)^2, na.rm = TRUE)/sum(num.i)
var.u
R.s.estimate.me(yone=d_example_me$y1, yzero=d_example_me$y0, sone=d_example_me$s1,
szero=d_example_me$s0, parametric = TRUE, estimator = "d", me.variance = var.u,
naive= TRUE, Ronly = FALSE)

R.s.estimate.me(yone=d_example_me$y1, yzero=d_example_me$y0,
sone=d_example_me$s1, szero=d_example_me$s0, parametric = FALSE, estimator = "q",
me.variance = 0.5, naive= FALSE, Ronly = TRUE)
```

`R.s.surv.estimate` *Calculates the proportion of treatment effect explained by the surrogate marker information measured at a specified time and primary outcome information up to that specified time*

Description

This function calculates the proportion of treatment effect on the primary outcome explained by the surrogate marker information measured at t_0 and primary outcome information up to t_0. The user can also request a variance estimate, estimated using perturbating-resampling, and a 95% confidence interval. If a confidence interval is requested three versions are provided: a normal approximation based interval, a quantile based interval and Fieller's confidence interval, all using perturbation-resampling. The user can also request an estimate of the incremental value of surrogate marker information.

Usage

```
R.s.surv.estimate(xone, xzero, deltaone, deltazero, sone, szero, t,
weight.perturb = NULL, landmark, extrapolate = FALSE, transform = FALSE,
conf.int = FALSE, var = FALSE, incremental.value = FALSE, approx = T)
```

Arguments

`xone`	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
`xzero`	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.

deltaone	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
deltazero	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
sone	numeric vector; surrogate marker measurement at t_0 for treated observations, assumed to be continuous. If $X_{1i} < t_0$, then the surrogate marker measurement should be NA.
szero	numeric vector; surrogate marker measurement at t_0 for control observations, assumed to be continuous. If $X_{1i} < t_0$, then the surrogate marker measurement should be NA.
t	the time of interest.
weight.perturb	weights used for perturbation resampling.
landmark	the landmark time t_0 or time of surrogate marker measurement.
extrapolate	TRUE or FALSE; indicates whether the user wants to use extrapolation.
transform	TRUE or FALSE; indicates whether the user wants to use a transformation for the surrogate marker.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval for delta is requested, default is FALSE.
var	TRUE or FALSE; indicates whether a variance estimate is requested, default is FALSE.
incremental.value	TRUE or FALSE; indicates whether the user would like to see the incremental value of the surrogate marker information, default is FALSE.
approx	TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

Details

Let G be the binary treatment indicator with $G = 1$ for treatment and $G = 0$ for control and we assume throughout that subjects are randomly assigned to a treatment group at baseline. Let $T^{(1)}$ and $T^{(0)}$ denote the time of the primary outcome of interest, death for example, under the treatment and under the control, respectively. Let $S^{(1)}$ and $S^{(0)}$ denote the surrogate marker measured at time t_0 under the treatment and the control, respectively.

The residual treatment effect is defined as

$$\Delta_S(t, t_0) = P(T^{(0)} > t_0) \left\{ \int \psi_1(t|s, t_0) dF_0(s|t_0) - P(T^{(0)} > t|T^{(0)} > t_0) \right\}$$

where $F_0(\cdot|t_0)$ is the cumulative distribution function of $S^{(0)}$ conditional on $T^{(0)} > t_0$ and $\psi_1(t|s, t_0) = P(T^{(1)} > t|S^{(1)} = s, T^{(1)} > t_0)$. The proportion of treatment effect explained by the surrogate marker information measured at t_0 and primary outcome information up to t_0, which we denote by $R_S(t, t_0)$, can be expressed using a contrast between $\Delta_S(t, t_0)$ and $\Delta(t)$:

$$R_S(t, t_0) = \{\Delta(t) - \Delta_S(t, t_0)\}/\Delta(t) = 1 - \Delta_S(t, t_0)/\Delta(t).$$

The definition and estimation of $\Delta(t)$ is described in the delta.surv.estimate documentation.

Due to censoring, our data consist of n_1 observations $\{(X_{1i}, \delta_{1i}, S_{1i}), i = 1, ..., n_1\}$ from the treatment group $G = 1$ and n_0 observations $\{(X_{0i}, \delta_{0i}, S_{0i}), i = 1, ..., n_0\}$ from the control group $G = 0$ where $X_{gi} = \min(T_{gi}, C_{gi})$, $\delta_{gi} = I(T_{gi} < C_{gi})$, C_{gi} denotes the censoring time, and S_{gi} denotes the surrogate marker information measured at time t_0, for $g = 1, 0$, for individual i. Note that if $X_{gi} < t_0$, then S_{gi} should be NA (not available).

To estimate $\Delta_S(t, t_0)$, we use a nonparametric kernel Nelson-Aalen estimator to estimate $\psi_1(t|s, t_0)$ as $\hat{\psi}_1(t|s, t_0) = \exp\{-\hat{\Lambda}_1(t|s, t_0)\}$, where

$$\hat{\Lambda}_1(t|s, t_0) = \int_{t_0}^{t} \frac{\sum_{i=1}^{n_1} I(X_{1i} > t_0) K_h\{\gamma(S_{1i}) - \gamma(s)\} dN_{1i}(z)}{\sum_{i=1}^{n_1} I(X_{1i} > t_0) K_h\{\gamma(S_{1i}) - \gamma(s)\} Y_{1i}(z)},$$

is a consistent estimate of $\Lambda_1(t|s, t_0) = -\log[\psi_1(t|s, t_0)], Y_{1i}(t) = I(X_{1i} \geq t), N_{1i}(t) = I(X_{1i} \leq t)\delta_i, K(\cdot)$ is a smooth symmetric density function, $K_h(x) = K(x/h)/h, \gamma(\cdot)$ is a given monotone transformation function, and h is a specified bandwidth. To obtain an appropriate h we first use bw.nrd to obtain h_{opt}; and then we let $h = h_{opt} n_1^{-c_0}$ with $c_0 = 0.11$.

Since $F_0(s|t_0) = P(S_{0i} \leq s|X_{0i} > t_0)$, we empirically estimate $F_0(s|t_0)$ using all subjects with $X_{0i} > t_0$ as

$$\hat{F}_0(s|t_0) = \frac{\sum_{i=1}^{n_0} I(S_{0i} \leq s, X_{0i} > t_0)}{\sum_{i=1}^{n_0} I(X_{0i} > t_0)}.$$

Subsequently, we construct an estimator for $\Delta_S(t, t_0)$ as

$$\hat{\Delta}_S(t, t_0) = n_0^{-1} \sum_{i=1}^{n_0} \left[\hat{\psi}_1(t|S_{0i}, t_0) \frac{I(X_{0i} > t_0)}{\hat{W}_0^C(t_0)} - \frac{I(X_{0i} > t)}{\hat{W}_0^C(t)} \right]$$

where $\hat{W}_g^C(\cdot)$ is the Kaplan-Meier estimator of survival for censoring for $g = 1, 0$. Finally, we estimate $R_S(t, t_0)$ as $\hat{R}_S(t, t_0) = 1 - \hat{\Delta}_S(t, t_0)/\hat{\Delta}(t)$.

Variance estimation and confidence interval construction are performed using perturbation-resampling. Specifically, let $\left\{ V^{(b)} = (V_{11}^{(b)}, ...V_{1n_1}^{(b)}, V_{01}^{(b)}, ...V_{0n_0}^{(b)})^T, b = 1, ..., D \right\}$ be $n \times D$ independent copies of a positive random variables V from a known distribution with unit mean and unit variance. Let

$$\hat{\Delta}^{(b)}(t) = \frac{\sum_{i=1}^{n_1} V_{1i}^{(b)} I(X_{1i} > t)}{\sum_{i=1}^{n_1} V_{1i}^{(b)} \hat{W}_1^{C(b)}(t)} - \frac{\sum_{i=1}^{n_0} V_{0i}^{(b)} I(X_{0i} > t)}{\sum_{i=1}^{n_0} V_{0i}^{(b)} \hat{W}_0^{C(b)}(t)}.$$

In this package, we use weights generated from an Exponential(1) distribution and use $D = 500$. The variance of $\hat{\Delta}(t)$ is obtained as the empirical variance of $\{\hat{\Delta}(t)^{(b)}, b = 1, ..., D\}$. Variance estimates for $\hat{\Delta}_S(t, t_0)$ and $\hat{R}_S(t, t_0)$ are calculated similarly. We construct two versions of the 95% confidence interval for each estimate: one based on a normal approximation confidence interval using the estimated variance and another taking the 2.5th and 97.5th empirical percentile of the perturbed quantities. In addition, we use Fieller's method to obtain a third confidence interval for $R_S(t, t_0)$ as

$$\left\{ 1 - r : \frac{(\hat{\Delta}_S(t, t_0) - r\hat{\Delta}(t))^2}{\hat{\sigma}_{11} - 2r\hat{\sigma}_{12} + r^2\hat{\sigma}_{22}} \leq c_\alpha \right\},$$

where $\hat{\Sigma} = (\hat{\sigma}_{ij})_{1 \leq i,j \leq 2}$ and c_α is the $(1 - \alpha)$th percentile of

$$\left\{ \frac{\{\hat{\Delta}_S^{(b)}(t) - (1 - \hat{R}_S(t, t_0))\hat{\Delta}(t)^{(b)}\}^2}{\hat{\sigma}_{11} - 2(1 - \hat{R}_S(t, t_0))\hat{\sigma}_{12} + (1 - \hat{R}_S(t, t_0))^2\hat{\sigma}_{22}}, b = 1, \cdots, C \right\}$$

where $\alpha = 0.05$.

Since the definition of $R_S(t, t_0)$ considers the surrogate information as a combination of both S information and T information up to t_0, a logical inquiry would be how to assess the incremental value of the S information in terms of the proportion of treatment effect explained, when added to T information up to t_0. The proportion of treatment effect explained by T information up to t_0 only is denoted as $R_T(t, t_0)$ and is described in the documentation for R.t.surv.estimate. The incremental value of S information is defined as:

$$IV_S(t, t_0) = R_S(t, t_0) - R_T(t, t_0) = \frac{\Delta_T(t, t_0) - \Delta_S(t, t_0)}{\Delta(t)}.$$

For estimation of $R_T(t_0)$, see documentation for R.t.surv.estimate. The quantity $IV_S(t, t_0)$ is then estimated by $\hat{IV}_S(t, t_0) = \hat{R}_S(t, t_0) - \hat{R}_T(t, t_0)$. Perturbation-resampling is used for variance estimation and confidence interval construction for this quantity, similar to the other quantities in this package.

Note that if the observed supports for S are not the same, then $\hat{\Lambda}_1(t|s, t_0)$ for $S_{0i} = s$ outside the support of S_{1i} may return NA (depending on the bandwidth). If extrapolation = TRUE, then the $\hat{\Lambda}_1(t|s, t_0)$ values for these surrogate values are set to the closest non-NA value. If transform = TRUE, then S_{1i} and S_{0i} are transformed such that the new transformed values, S_{1i}^{tr} and S_{0i}^{tr} are defined as: $S_{gi}^{tr} = F([S_{gi} - \mu]/\sigma)$ for $g = 0, 1$ where $F(\cdot)$ is the cumulative distribution function for a standard normal random variable, and μ and σ are the sample mean and standard deviation, respectively, of $\{(S_{1i}, S_{0i})^T, i \quad s.t. X_{gi} > t_0\}$.

Value

A list is returned:

`delta` the estimate, $\hat{\Delta}(t)$, described in delta.estimate documentation.

`delta.s` the estimate, $\hat{\Delta}_S(t, t_0)$, described above.

`R.s` the estimate, $\hat{R}_S(t, t_0)$, described above.

`delta.var` the variance estimate of $\hat{\Delta}(t)$; if var = TRUE or conf.int = TRUE.

`delta.s.var` the variance estimate of $\hat{\Delta}_S(t, t_0)$; if var = TRUE or conf.int = TRUE.

`R.s.var` the variance estimate of $\hat{R}_S(t, t_0)$; if var = TRUE or 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.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.

`conf.int.normal.delta.s`
a vector of size 2; the 95% confidence interval for $\hat{\Delta}_S(t, t_0)$ based on a normal approximation; if conf.int = TRUE.

`conf.int.quantile.delta.s`
a vector of size 2; the 95% confidence interval for $\hat{\Delta}_S(t, t_0)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE.

`conf.int.normal.R.s`
: a vector of size 2; the 95% confidence interval for $\hat{R}_S(t, t_0)$ based on a normal approximation; if conf.int = TRUE.

`conf.int.quantile.R.s`
: a vector of size 2; the 95% confidence interval for $\hat{R}_S(t, t_0)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE.

`conf.int.fieller.R.s`
: a vector of size 2; the 95% confidence interval for $\hat{R}_S(t, t_0)$ based on Fieller's approach, described above; if conf.int = TRUE.

`delta.t`
: the estimate, $\hat{\Delta}_T(t, t_0)$, described above; if incremental.vaue = TRUE.

`R.t`
: the estimate, $\hat{R}_T(t, t_0)$, described above; if incremental.vaue = TRUE.

`incremental.value`
: the estimate, $I\hat{V}_S(t, t_0)$, described above; if incremental.vaue = TRUE.

`delta.t.var`
: the variance estimate of $\hat{\Delta}_T(t, t_0)$; if var = TRUE or conf.int = TRUE and incremental.vaue = TRUE.

`R.t.var`
: the variance estimate of $\hat{R}_T(t, t_0)$; if var = TRUE or conf.int = TRUE and incremental.vaue = TRUE.

`incremental.value.var`
: the variance estimate of $I\hat{V}_S(t, t_0)$; if var = TRUE or conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.normal.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t, t_0)$ based on a normal approximation; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.quantile.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t, t_0)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.normal.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on a normal approximation; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.quantile.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.fieller.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on Fieller's approach, described above; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.normal.iv`
: a vector of size 2; the 95% confidence interval for $I\hat{V}_S(t, t_0)$ based on a normal approximation; if conf.int = TRUE and incremental.vaue = TRUE.

`conf.int.quantile.iv`
: a vector of size 2; the 95% confidence interval for $I\hat{V}_S(t, t_0)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE and incremental.vaue = TRUE.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting". If the observed support of the surrogate marker for the control group is outside the observed support of the surrogate marker for the treatment group, the user will receive the following message: "Warning: observed supports do not appear equal, may need to consider a transformation or extrapolation".

Author(s)

Layla Parast

References

Parast, L., Cai, T., & Tian, L. (2017). Evaluating surrogate marker information using censored data. Statistics in Medicine, 36(11), 1767-1782.

Examples

```
data(d_example_surv)
names(d_example_surv)
```

R.t.surv.estimate *Calculates the proportion of treatment effect explained by the primary outcome information up to a specified time*

Description

This function calculates the proportion of treatment effect on the primary outcome explained by the treatment effect on the primary outcome up to t_0. The user can also request a variance estimate, estimated using perturbating-resampling, and a 95% confidence interval. If a confidence interval is requested three versions are provided: a normal approximation based interval, a quantile based interval and Fieller's confidence interval, all using perturbation-resampling.

Usage

```
R.t.surv.estimate(xone, xzero, deltaone, deltazero, t, weight.perturb = NULL,
landmark, var = FALSE, conf.int = FALSE, approx = T)
```

Arguments

xone	numeric vector, the observed event times in the treatment group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
xzero	numeric vector, the observed event times in the control group, X = min(T,C) where T is the time of the primary outcome and C is the censoring time.
deltaone	numeric vector, the event indicators for the treatment group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.

deltazero	numeric vector, the event indicators for the control group, D = I(T<C) where T is the time of the primary outcome and C is the censoring time.
t	the time of interest.
weight.perturb	weights used for perturbation resampling.
landmark	the landmark time t_0 or time of surrogate marker measurement.
var	TRUE or FALSE; indicates whether a variance estimate for delta is requested, default is FALSE.
conf.int	TRUE or FALSE; indicates whether a 95% confidence interval for delta is requested, default is FALSE.
approx	TRUE or FALSE indicating whether an approximation should be used when calculating the probability of censoring; most relevant in settings where the survival time of interest for the primary outcome is greater than the last observed event but before the last censored case, default is TRUE.

Details

Let G be the binary treatment indicator with $G = 1$ for treatment and $G = 0$ for control and we assume throughout that subjects are randomly assigned to a treatment group at baseline. Let T denote the time of the primary outcome of interest, death for example. We use potential outcomes notation such that $T^{(g)}$ denotes the time of the primary outcome under treatment $G = g$. The proportion of treatment effect explained by T observed up to t_0 only is $R_T(t, t_0) = 1 - \Delta_T(t, t_0)/\Delta(t)$ where

$$\Delta_T(t, t_0) = P(T^{(0)} > t_0)P(T^{(1)} > t \mid T^{(1)} > t_0) - P(T^{(0)} > t).$$

To estimate $R_T(t, t_0)$, we use the estimator $\hat{R}_T(t, t_0) = 1 - \hat{\Delta}_T(t, t_0)/\hat{\Delta}(t)$ where $\hat{\Delta}_T(t, t_0) = \hat{\phi}_0(t_0)\hat{\phi}_1(t)/\hat{\phi}_1(t_0) - \hat{\phi}_0(t)$ and $\hat{\phi}_g(u) = n_g^{-1} \sum_{i=1}^{n_g} \frac{I(X_{gi} > u)}{\hat{W}_g^C(u)}$ for $g = 1, 0$ where $\widehat{W}_g^C(\cdot)$ is the Kaplan-Meier estimator of survival for censoring for $g = 1, 0$.

Value

A list is returned:

delta	the estimate, $\hat{\Delta}(t)$, described in delta.estimate documentation.
delta.t	the estimate, $\hat{\Delta}_T(t, t_0)$, described above.
R.t	the estimate, $\hat{R}_T(t, t_0)$, described above.
delta.var	the variance estimate of $\hat{\Delta}(t)$; if var = TRUE or conf.int = TRUE.
delta.t.var	the variance estimate of $\hat{\Delta}_T(t, t_0)$; if var = TRUE or conf.int = TRUE.
R.t.var	the variance estimate of $\hat{R}_T(t, t_0)$; if var = TRUE or 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.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.

`conf.int.normal.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t, t_0)$ based on a normal approximation; if conf.int = TRUE.

`conf.int.quantile.delta.t`
: a vector of size 2; the 95% confidence interval for $\hat{\Delta}_T(t, t_0)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE.

`conf.int.normal.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on a normal approximation; if conf.int = TRUE.

`conf.int.quantile.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on sample quantiles of the perturbed values, described above; if conf.int = TRUE.

`conf.int.fieller.R.t`
: a vector of size 2; the 95% confidence interval for $\hat{R}_T(t, t_0)$ based on Fieller's approach, described above; if conf.int = TRUE.

Note

If the treatment effect is not significant, the user will receive the following message: "Warning: it looks like the treatment effect is not significant; may be difficult to interpret the residual treatment effect in this setting".

Author(s)

Layla Parast

References

Parast, L., Cai, T., & Tian, L. (2017). Evaluating surrogate marker information using censored data. Statistics in Medicine, 36(11), 1767-1782.

Examples

```
data(d_example_surv)
names(d_example_surv)
```

Index

∗ **augment**
Aug.R.s.surv.estimate, 2
∗ **datasets**
d_example, 17
d_example_me, 18
d_example_multiple, 18
d_example_surv, 19
∗ **htest**
fieller.ci, 20
me.variance.estimate, 21
∗ **models**
delta.s.estimate, 10
R.s.estimate, 28
R.s.estimate.me, 33
∗ **nonparametric**
Aug.R.s.surv.estimate, 2
delta.multiple.surv, 8
delta.s.estimate, 10
delta.s.surv.estimate, 12
delta.surv.estimate, 13
delta.t.surv.estimate, 16
R.multiple.surv, 22
R.s.estimate, 28
R.s.estimate.me, 33
R.s.surv.estimate, 39
R.t.surv.estimate, 44
∗ **regression**
delta.s.estimate, 10
R.s.estimate, 28
R.s.estimate.me, 33
∗ **robust**
Aug.R.s.surv.estimate, 2
delta.multiple.surv, 8
delta.s.estimate, 10
delta.s.surv.estimate, 12
delta.t.surv.estimate, 16
R.multiple.surv, 22
R.s.estimate, 28
R.s.estimate.me, 33
R.s.surv.estimate, 39
R.t.surv.estimate, 44
∗ **smooth**
Aug.R.s.surv.estimate, 2
delta.multiple.surv, 8
delta.s.estimate, 10
delta.s.surv.estimate, 12
R.multiple.surv, 22
R.s.estimate, 28
R.s.estimate.me, 33
R.s.surv.estimate, 39
∗ **survival**
Aug.R.s.surv.estimate, 2
delta.multiple.surv, 8
delta.s.surv.estimate, 12
delta.surv.estimate, 13
delta.t.surv.estimate, 16
R.multiple.surv, 22
R.s.surv.estimate, 39
R.t.surv.estimate, 44
∗ **univar**
delta.estimate, 6
delta.surv.estimate, 13
me.variance.estimate, 21

Aug.R.s.surv.estimate, 2

d_example, 17
d_example_me, 18
d_example_multiple, 18
d_example_surv, 19
delta.estimate, 6
delta.multiple.surv, 8
delta.s.estimate, 10
delta.s.surv.estimate, 12
delta.surv.estimate, 13
delta.t.surv.estimate, 16

fieller.ci, 20

me.variance.estimate, 21

R.multiple.surv, 22
R.s.estimate, 28
R.s.estimate.me, 33
R.s.surv.estimate, 39
R.t.surv.estimate, 44