Package 'surviv'

May 9, 2026

Type Package

Title Flexible Instrumental Variable Methods for Survival Analysis

Version 0.1.3

Description Instrumental variable analysis methods for causal inference on survival data based on the Cox model allowing for various treatment and effect types, including orthogonality method-of-moments instrumental variable estimation for the Cox model, two-stage residual inclusion Cox estimation with frailty, sequential trial emulation, sequential Cox analyses, and sequential two-stage residual inclusion Cox analyses. Methodological background includes MacKenzie et al. (2014) <doi:10.1007/s10742-014-0117-x>, Martinez-Camblor et al. (2019) <doi:10.1093/biostatistics/kxx062>, Martinez-Camblor et al. (2019) <doi:10.1111/rssc.12341>, Gran et al. (2010) <doi:10.1002/sim.4048>, and Keogh et al. (2023) <doi:1

License MIT + file LICENSE

Depends R (>= 2.10)

Imports MASS, survival (>= 3.4-0), rootSolve

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

URL https://github.com/tonyhbc/surviv

BugReports https://github.com/tonyhbc/surviv/issues

NeedsCompilation no

Author Haobin Chen [aut, cre],
James O'Malley [aut],
Pablo Martinez-Camblor [aut],
Todd MacKenzie [aut]

Maintainer Haobin Chen <tony.haobin.chen@alumni.emory.edu>

Repository CRAN

Date/Publication 2026-04-24 18:30:02 UTC

Contents

`coxiv_omom` *Orthogonality method-of-moment IV estimation of Cox model*

Description

Estimate marginal causal treatment effect (in terms of log-hazard ratio) in Cox model setting subject to unmeasured confounding U using instrumental variable based on the IV **O**rthogonality **M**ethod-**o**f-**M**oment estimator in MacKenzie et al. (2014).

Usage

```
coxiv_omom(formula, trt, iv, data)
```

Arguments

`formula`	a regression formula object with Surv object as outcome including survival time and censoring indicator.
`trt`	a string indicates the name of the treatment variable of interest. Only time-invariant **binary** or **continuous** treatment is supported. It is strongly recommended that binary treatment is converted to numeric type (0,1) instead of using factor type. Treatment cannot be multi-categorical.
`iv`	a string indicates the name of the instrumental variable. Only one time-variant **binary** or **continuous** instrumental variable is supported. IV cannot be multi-categorical.
`data`	a data.frame object of input data containing all variables specified in `formula`, including observed confounders.

Details

A causal Cox counterfactual survival model for survival time T with additive unmeasured confounding is assumed for treatment A, observed confounder X, and unmeasured confounder U.

$$P(T(a) = t|T(a) \geq t, U = u) = \lambda_0(t)\exp(\beta_a a + \beta' x) + h(u, t)$$

OMOM IV estimator assumes the following conditions: 1) relevance $Z \not\perp\!\!\!\perp A|U, X$, 2) exclusion restriction $Z \perp\!\!\!\perp (T, D)|A, U, X$, and 3) randomization $Z \perp\!\!\!\perp U|X$. The estimates are obtained as the numerical solution to zero-crossing of estimating equation:

$$\psi(\beta) \approx \sum_{i=1}^{n} \int_0^{\tau} \left\{ Z_i - \frac{\sum_{j=1}^{n} Z_j D_j(s)\exp\{\beta_a A_j + \beta' X\}}{\sum_{j=1}^{n} D_j(s)\exp\{\beta_a A_j + \beta' X\}} dN_i(s) \right\} = 0$$

which is identical to the partial Cox score function for instrument Z. $D(t)$ is the event indicator at time t and $N(t)$ is a counting process for survival. Huber-White sandwich estimator is used to obtain variance for $\hat{\beta}$. Suggestive diagnostic test for weak IV was provided by fitting a pseudo-first stage treatment linear model (since OMOM IV is an estimating equation based IV approach, not two-stage based) regressing treatment `trt` on IV `iv` and confounders extracted from all predictors from `formula` input except for `trt`. A nested anova F-test was performed to compare fitted treatment models with and without `iv` as predictor. A popular rule of thumb of indication for weak IV in econometric literature is when F-statistics is less than 10 from comparing IV-exclusion model.

Value

A `coxivomom` object for OMOM IV estimation of a Cox model with the following components:

- `call`: the match call of the OMOM IV model.
- `coef` : a `data.frame` of estimated log-hazard ratios and inference for all predictors in the outcome Cox model, with columns `logHR`, `se`, `Z`, and `p.val`.
- `conf_int` : a `data.frame` of 95% confidence interval(s) for the estimated log hazard ratios.
- `input`: the collected input of the OMOM IV model.
- `formula`: the outcome regression formula for Cox model
- `n`: the effective sample size used for model estimation after missingness removal.
- `events`: the effective number of events of survival outcome after missingness removal.
- `vcov_mat` : estimated covariance matrix of log hazard ratios.
- `iv_diag`: a list of IV diagnosis for weak IV. Include F-statistics and `anova` fit of nested treatment models comparison.
- `surv_curve`: a survfit object for fitted survival curve using complete data.

References

MacKenzie, T.A., Tosteson, T.D., Morden, N.E. et al. Using instrumental variables to estimate a Cox proportional hazards regression subject to additive confounding. *Health Serv Outcomes Res Method* 14, 54-68 (2014).

Examples

```
data("VitD", package = "surviv")
fit_omom <- coxiv_omom(survival::Surv(time, death) ~ vitd + age,
                       trt = "vitd", iv = "filaggrin", data = VitD)
fit_omom$coef
fit_omom$conf_int
```

`coxiv_seq` *Sequential 2SRI Cox analysis via sequential trials emulation*

Description

Fits a sequential two-stage residual inclusion (2SRI) Cox model for a time-varying treatment with a baseline instrumental variable, using a stacked sequential trial emulation created by `seqem()`.

Usage

```
coxiv_seq(formula, trtformula, data, id, tvtrt, iv, stratify = FALSE,
  se_type = c("robust", "jackknife", "bootstrap", "none"), B = 50,
  seed = NULL, ipacw_den_covs = NULL, ipacw_num_covs = NULL,
  by_trial = TRUE, ...)
```

Arguments

`formula`	a `formula` for the composite second-stage Cox model, typically of the form `Surv(start.new, stop.new, event) ~ tvtrt.base + x1.base + x2.base`. The right-hand side must include `paste0(tvtrt, ".base")`. The control function term `R.base` is added automatically if absent.
`trtformula`	a `formula` for the trial-specific first-stage treatment model. Its left-hand side must equal `paste0(tvtrt, ".base")`, and its right-hand side must include `paste0(iv, ".base")`.
`data`	a `seqem` object returned by `seqem()`.
`id`	a string name for the subject identifier variable.
`tvtrt`	a string name for the original time-varying treatment variable.
`iv`	a string name for the original baseline instrumental variable.
`stratify`	logical; if `TRUE`, the composite Cox model is stratified by emulated trial so that each trial has its own baseline hazard.
`se_type`	standard error estimator: one of `"robust"`, `"jackknife"`, `"bootstrap"`, or `"none"`.
`B`	the number of bootstrap replicates when `se_type = "bootstrap"`.
`seed`	optional random seed for bootstrap resampling.

ipacw_den_covs	optional string vector of underlying covariate names to use in the denominator IPACW model through their lead versions (for example c("endoleak") uses endoleak.lead1). If NULL, the underlying variables corresponding to non-treatment .base terms in formula and trtformula are used whenever the original variables are available in the stacked data.
ipacw_num_covs	optional string vector of baseline .base covariates to use in the numerator stabiliser model. If NULL, an intercept-only numerator model is used.
by_trial	logical; if TRUE, fit and return separate second-stage Cox models within each emulated trial.
...	additional arguments passed to survival::coxph().

Details

The analysis proceeds trial by trial. Within each emulated trial, a first-stage logistic regression is fit for the trial-baseline treatment {tvtrt}.base on the baseline IV and baseline covariates from trtformula. A trial-specific control-function residual R.base is then constructed and carried across all rows within that subject-trial trajectory.

The stacked data are then artificially censored when observed treatment deviates from the trial-baseline treatment status, and stabilised inverse probability of artificial censoring weights (IPACW) are estimated. These weights address the informative censoring induced by the sequential trial emulation itself. This function does not estimate additional weights for naturally occurring right censoring.

Finally, a weighted composite Cox model is fit on the artificially censored, weighted stacked data. The returned coefficient summary is organised in the same style as seqcox(), coxiv_omom(), and coxiv_tsrif().

Value

A coxivseq object with components including:

- call : the matched call.
- formula : the composite second-stage Cox formula actually fitted.
- trtformula : the sequential first-stage treatment models formula.
- fit : the fitted composite survival::coxph() model.
- coef : a data.frame with columns logHR, HR, se, Z, p.val.
- conf_int : a data.frame of 95% confidence limits with columns 2.5% and 97.5% for estimated log hazard ratios.
- vcov_mat : variance-covariance matrix corresponding to the selected se_type.
- trtfit_by_trial : list of first-stage treatment models.
- fit_by_trial : list of second-stage trial-specific Cox fits when by_trial = TRUE.
- data_ste : final stacked, artificially censored analysis dataset, including R.base and the computed IPACW column.
- jackknife, bootstrap: a matrix of jackknife or bootstrap resampled estimates if se_type set to jackknife or boostrap.
- tvtrt, iv, n, n_ids, n_trials, n_event, stratify, se_type, B, by_trial: miscellaneous string tags for model print summary.

References

Gran JM, Roysland K, Wolbers M, Didelez V, Sterne JA, Ledergerber B, Furrer H, von Wyl V, Aalen OO. A sequential Cox approach for estimating the causal effect of treatment in the presence of time-dependent confounding applied to data from the Swiss HIV Cohort Study. *Statistics in Medicine*. 2010;29(26):2757-2768.

Keogh RH, Gran JM, Seaman SR, Davies G, Vansteelandt S. Causal inference in survival analysis using longitudinal observational data: Sequential trials and marginal structural models. *Statistics in Medicine*. 2023;42(13):2191-2225.

Examples

```
data("vascular", package = "surviv")

vasc_seqem <- seqem(
  data = vascular,
  start = "time.start",
  stop = "time.stop",
  event = "event",
  id = "id",
  tvtrt = "reint",
  covs = c("iv", "diameter", "endoleak"),
  coarsen = "ceiling",
  cbin_width = 2
)

fit_seqcox <- coxiv_seq(
  formula = survival::Surv(start.new, stop.new, event) ~
    reint.base + diameter.base + endoleak.base,
  trtformula = reint.base ~ iv.base + diameter.base + endoleak.base,
  data = vasc_seqem,
  id = "id",
  tvtrt = "reint",
  iv = "iv",
  se_type = "robust",
  by_trial = TRUE
)

print(fit_seqcox)
```

`coxiv_tsrif` *Two-stage residual inclusion-frailty (TSRI-F) instrumental variable analysis of Cox model*

Description

Flexible instrumental analysis of Cox model in a novel **T**wo-**S**tage **R**esidual **I**nclusion with **F**railty (TSRI-F) framework when treatment A and mortality D are subject to unmeasured confounding.

Allow estimation of time constant treatment effect (Martinez-Camblor et al. (2019)) and time-varying treatment effect (Martinez-Camblor et al. (2019)) in terms of log hazard ratio (log-HR). A set of instrumental variables Z and covariates for adjustment X (i.e., measured confounders) are required to estimate treatment effects $\beta_a^{(1)}$ and $\beta_a^{(2)}$ before and after the pre-specified time `tchange`.

Usage

```
coxiv_tsrif(surv, cens, trt, iv, covs, data, tchange = NULL,
            tvareff = FALSE, fdist = "gamma", bootvar = FALSE, B = 50)
```

Arguments

`surv`	a string indicating variable name of survival time T.
`cens`	a string indicating variable name of event indicator D: 1 for terminal event occurrence and 0 for right censor.
`trt`	a string indicating the treatment variable name A. Binary or continuous treatment supported.
`iv`	a string or vector of strings indicating instrumental variable(s) Z. Binary or continuous IV supported.
`covs`	a string or vector of strings indicating set of measured covariates X.
`data`	a data.frame containing all variables needed.
`tchange`	a positive numeric value specifying the time point at which the treatment effect changes value.
`tvareff`	logical; by default `FALSE` and estimate a time-constant treatment effect model. If `TRUE`, a time-varying treatment effect model is applied.
`fdist`	the frailty distribution, default is `gamma`. Other options include `gaussian` or `t` distribution. Read more about frailty distribution specification at frailty.
`bootvar`	logical; if `TRUE`, bootstrap variance estimation is performed. Otherwise, analytical variance is provided from frailty Cox model.
`B`	number of bootstrap iterations if `bootvar` is `TRUE`.

Details

This function performs two-stage residual inclusion IV analysis of the Cox model with individual frailty when the treatment effect β_a is constant over time (`tvareff = FALSE`) or $\beta_a(t)$ changes value at some time *t* (`tvareff = TRUE`). The IV assumptions are that 1) relevance $Z \not\perp\!\!\!\perp A|U, X$, 2) exclusion restriction $Z \perp\!\!\!\perp (T, D)|A, U, X$, and 3) randomization $Z \perp\!\!\!\perp U|X$ with U the unmeasured confounder. TSRI-F posits a first-stage linear model that generates the treatment:

$$A = \gamma_0 + \gamma_1 Z + \gamma_2 U + \gamma' X + \epsilon$$

and a second-stage outcome model follows Cox proportional hazards form:

$$\lambda(t|X, U) = \lambda_0(t) \exp\{\beta_a A + \beta_u U + \beta' X\}$$

where λ_0 is the baseline hazard function. The first stage of TSRIF procedure for constant treatment effect estimates *control function* $\widehat{R}$ from residual of treatment model: $\widehat{R} = A - \hat{\gamma}_0 - \hat{\gamma}_1 Z - \hat{\gamma}' X$, and fit a second-stage Cox model with user-specified individual frailty ϕ while adjusting for $\widehat{R}$.

For more intricate case of time-varying effect, the first stage proceeds identically to compute $\widehat{R}^{(1)} = A - \hat{\gamma}_0 - \hat{\gamma}_1 Z - \hat{\gamma}' X$. Second stage involves fitting two Cox models to estimate $\beta_a^{(1)} = \beta_a(t), t <$ `tchange` and $\beta_a^{(2)} = \beta_a(t), t \geq$ `tchange`. The first-period Cox model is fitted with frailty term $\hat{\phi}_1$ and including $\widehat{R}^{(1)}$ as a covariate. Then, update the control function $\widehat{R}^{(2)} = \widehat{R}^{(1)} + \hat{\beta}_r^{(1),-1}\hat{\phi}_1$ for the second period, which are then included as a covariate in the second-period Cox model with a new frailty term.

This approach estimates conditional treatment effect of `trt`. Coefficient estimates of covariates other than `trt` are also estimated and readily available, but these estimates are suggestive of their associational relations with survival outcome, and causal conclusions regarding these covariates with outcome should be cautious.

Value

A `coxivtsrif` object with following components:

- `input` : collected input and derived model formulas used for the TSRI-F fit.
- `trt_coef` : a `data.frame` of treatment-effect estimates with columns `logHR`, `se`, `Z`, and `p.val`. If `tvareff = TRUE`, it contains one row for each treatment-effect period.
- `conf_int` : a `data.frame` for 95% confidence interval(s) for estimated treatment effect (s) from the choice of standard error estimator.
- `ctrl_func` : the estimated control function $\widehat{R}$, or `data.frame` of $\widehat{R}^1$ and $\widehat{R}^2$ if `tvareff = TRUE`.
- `trt_model` : a lm object fit of the first-stage treatment model.
- `out_model` : a coxph object fit of the second-stage outcome Cox model. If `tvareff = TRUE`, Cox models for first period `out_model1` and second period `out_model2` are both returned.
- `est_boot` : a vector of length `B` of bootstrap estimates from each iteration. If `bootvar = FALSE`, returns `NULL`.
- `data_long` : a `data.frame` of data for model fit transformed into start-stop form for two-period Cox estimation, if `tvareff = TRUE`.
- `surv_curve`: a survfit object for fitted survival curve using complete data.
- `iv_diag` : a list of IV diagnostics for weak-IV assessment, including the nested ANOVA comparison and F-statistic.

References

1. Pablo Martinez-Camblor, Todd Mackenzie, Douglas O Staiger, Philip P Goodney, A James O'Malley, Adjusting for bias introduced by instrumental variable estimation in the Cox proportional hazards model, *Biostatistics*, Volume 20, Issue 1, January 2019, Pages 80-96,
2. Pablo Martinez-Camblor, Todd A. MacKenzie, Douglas O. Staiger, Phillip P. Goodney, A. James O'Malley, An Instrumental Variable Procedure for Estimating Cox Models with Non-Proportional Hazards in the Presence Of Unmeasured Confounding, *Journal of the Royal Statistical Society Series C: Applied Statistics*, Volume 68, Issue 4, August 2019, Pages 985-1005

Examples

```
data("VitD", package = "surviv")

fit_tsrif1 <- coxiv_tsrif(surv = "time", cens = "death", covs = c("age"),
                          trt = "vitd", iv = "filaggrin", tchange = NULL,
                          data = VitD, fdist = "gaussian", bootvar = FALSE,
                          B = 20, tvareff = FALSE)
fit_tsrif1$trt_coef

fit_tsrif2 <- coxiv_tsrif(surv = "time", cens = "death", covs = c("age"),
                          trt = "vitd", iv = "filaggrin", tchange = 5,
                          data = VitD, fdist = "gaussian", bootvar = FALSE,
                          B = 20, tvareff = TRUE)
fit_tsrif2$trt_coef
```

---

print.coxivomom *Print the output from a* coxivomom *model fit.*

Description

Provide succinct and organized output for a coxivomom model, such as coxiv_omom().

Usage

```
## S3 method for class 'coxivomom'
print(x, digits = 3L, ...)
```

Arguments

x	an object of class "coxivomom", usually created by coxiv_omom().
digits	the number of decimal places to display for coefficient summaries and hazard ratios.
...	additional arguments (currently unused).

Value

The input object x, invisibly. Called for its side effect of printing a summary to the console.

Examples

```
data("VitD", package = "surviv")
fit_omom <- surviv::coxiv_omom(
  formula = survival::Surv(time, death) ~ vitd + age,
  trt = "vitd",
  iv = "filaggrin",
  data = VitD
)
print(fit_omom)
```

print.coxivseq *Print method for coxivseq objects*

Description

Print method for coxivseq objects

Usage

```
## S3 method for class 'coxivseq'
print(x, digits = 4, ...)
```

Arguments

x	An object of class coxivseq returned by coxiv_seq().
digits	Number of decimal places for printed coefficient summaries.
...	Ignored.

Value

The input object x, invisibly. Called for its side effect of printing a summary to the console.

print.coxivtsrif *Print the output from a* coxivtsrif *model fit.*

Description

Provide succinct and organized output for a coxivtsrif model, such as coxiv_tsrif().

Usage

```
## S3 method for class 'coxivtsrif'
print(x, digits = 3L, ...)
```

Arguments

x	an object of class "coxivtsrif", usually created by coxiv_tsrif().
digits	the number of decimal places to display for coefficient summaries and hazard ratios.
...	additional arguments (currently unused).

Value

The input object x, invisibly. Called for its side effect of printing a summary to the console.

Examples

```
data("VitD", package = "surviv")
fit_tsrif <- surviv::coxiv_tsrif(
  surv = "time",
  cens = "death",
  covs = c("age"),
  trt = "vitd",
  iv = "filaggrin",
  tchange = NULL,
  data = VitD,
  fdist = "gaussian",
  bootvar = FALSE,
  B = 20,
  tvareff = FALSE
)
print(fit_tsrif)
```

`print.seqcox` *Print method for seqcox objects*

Description

Print method for seqcox objects

Usage

```
## S3 method for class 'seqcox'
print(x, digits = 4, ...)
```

Arguments

`x`	An object of class `seqcox` returned by `seqcox()`.
`digits`	Number of decimal places for printed coefficient summaries.
`...`	Ignored.

Value

The input object `x`, invisibly. Called for its side effect of printing a summary to the console.

print.seqem *Print method for seqem objects*

Description

Print method for seqem objects

Usage

```
## S3 method for class 'seqem'
print(x, ...)
```

Arguments

x	An object of class seqem returned by seqem().
...	Ignored.

Value

The input object x, invisibly. Called for its side effect of printing a summary to the console.

seqcox *Sequential Cox analysis via sequential trials emulation*

Description

Fit a composite Cox model to a stacked sequential trial emulation object produced by seqem(). The analysis follows the sequential Cox approach of Gran et al. by using stabilized inverse probability of artificial censoring weights (IPACW) to adjust for the informative artificial censoring induced by restricting each emulated trial to individuals who continue to follow their treatment assignment at the trial start.

Usage

```
seqcox(formula, data, id, trial_id = "trial", stratify = FALSE,
  se_type = c("robust", "jackknife", "bootstrap", "none"), B = 50,
  seed = NULL, ipacw_trunc = 0.95, ipacw_den_covs = NULL,
  ipacw_num_covs = NULL, by_trial = TRUE, ...)
```

Arguments

formula	a survival formula for the composite Cox model, typically of the form `Surv(start.new, stop.new, event) ~ trt.base + L.base + ...`. The *first* `.base` term is treated as the trial-baseline treatment variable.
data	a `seqem` object returned by `seqem()`. A raw data frame is not accepted.
id	a string name of the subject identifier column in the sequential trials data.
trial_id	a string name of the trial index column. Defaults to `"trial"`.
stratify	logical; if `TRUE`, the composite Cox model is stratified by `trial_id` so each emulated trial has its own baseline hazard.
se_type	standard error estimator: one of `"robust"`, `"jackknife"`, `"bootstrap"`, or `"none"`.
B	number of bootstrap replicates when `se_type = "bootstrap"`.
seed	optional random seed for the bootstrap.
ipacw_trunc	optional truncation quantile for cumulative stabilized IPACW. Defaults to `0.95`. Set to `NULL` to disable truncation.
ipacw_den_covs	optional character vector of underlying covariate names to use in the denominator IPACW model through their lead versions (for example `c("endoleak")` uses `endoleak.lead1`). If `NULL`, the underlying variables corresponding to non-treatment `.base` terms in `formula` are used.
ipacw_num_covs	optional character vector of baseline `.base` covariates to use in the numerator stabilizer model. If `NULL`, an intercept-only numerator model is used.
by_trial	logical; if `TRUE` (default), fit trial-specific Cox models and return them in `$fit_by_trial`.
...	additional arguments passed to `survival::coxph()`.

Value

A `seqcox` object with components including:

- `call` : the matched call.
- `formula` : the composite Cox formula actually fitted.
- `fit` : a composite `survival::coxph()` fit.
- `coef` : a `data.frame` of estimated coefficient summaries with columns `logHR`, `HR`, `se`, `Z`, and `p.val`.
- `conf_int` : a `data.frame` of 95% confidence limits with columns `2.5%` and `97.5%`.
- `n_ids`, `n_trials`, `n_event`: the analysis size summaries for sample size, number of emulated trials aggregated, and number of events.
- `fit_by_trial` : a list of trial-specific Cox fits when `by_trial = TRUE`.
- `data_ste` : a `data.frame` of stacked analysis dataset actually used in the composite fit, including the computed IPACW column.
- `jackknife`, `bootstrap`: a vector of jackknife or bootstrap resampled estimates if `se_type` set to `jackknife` or `boostrap`.

References

Gran JM, Roysland K, Wolbers M, Didelez V, Sterne JA, Ledergerber B, Furrer H, von Wyl V, Aalen OO. A sequential Cox approach for estimating the causal effect of treatment in the presence of time-dependent confounding applied to data from the Swiss HIV Cohort Study. *Statistics in Medicine*. 2010;29(26):2757-2768.

Keogh RH, Gran JM, Seaman SR, Davies G, Vansteelandt S. Causal inference in survival analysis using longitudinal observational data: Sequential trials and marginal structural models. *Statistics in Medicine*. 2023;42(13):2191-2225.

Examples

```
data("vascular", package = "surviv")
vasc_seqem = seqem(data = vascular, start = "time.start",
                   stop = "time.stop", event = "event", id = "id",
                   tvtrt = "reint", covs = c("diameter", "endoleak"),
                   coarsen = "ceiling", cbin_width = 2)
fit_seqcox <- seqcox(
  formula = survival::Surv(start.new, stop.new, event) ~
    reint.base + diameter.base + endoleak.base,
  data = vasc_seqem,
  id = "id",
  se_type = "robust"
)
print(fit_seqcox)
```

`seqem` *Sequential trial emulation for time-dependent treatment data*

Description

Constructs a stacked sequence of emulated trials from longitudinal survival data in start-stop form, following the sequential trials framework of Gran et al. and Keogh et al.

Usage

```
seqem(data, start, stop, event, tvtrt, id, covs = NULL, coarsen = c("none",
  "floor", "ceiling"), cbin_width = NULL)
```

Arguments

`data`	a `data.frame` in start-stop long format with potentially multiple rows per individual. Typically a dataset produced by tmerge.
`start`	a string name of the interval start time variable.
`stop`	a string name of the stop time variable.
`event`	a string name of the event indicator (1: event).

tvtrt	a string name of the time-varying **treatment** variable. Currently this must be binary and monotone in the sense assumed by the sequential trials framework (remain 1 once become 1). A trial-specific baseline version of tvtrt is always created as tvtrt.base.
id	a string name of the unique individual identifier.
covs	optional. a string vector of covariate names in data for which trial-specific baseline versions should be created. Both *time-constant* and *time-varying* covariates are supported. For each X in covs, seqem() creates X.base, representing the value of X at the start of each emulated trial for each individual.
coarsen	one of "none", "floor", or "ceiling". "none" leaves the original timeline unchanged. "floor" and "ceiling" discretize row start times to the coarsening grid and thus affect both treatment and time-varying covariate histories.
cbin_width	a positive numeric bin width used when coarsen = "floor" or "ceiling".

Details

Trial start times are determined only by the earliest observed cohort entry time and by times at which at least one individual newly initiates treatment (0 -> 1). Changes in time-varying covariates listed in covs do **not** themselves create new trial start times. However, seqem() still carries those covariates forward correctly into each emulated trial by creating trial-specific baseline versions with suffix .base.

When coarsen != "none", the observed timeline is discretized/coarsened onto a grid of time width cbin_width. This coarsening is applied to **all** row start times, not only to treatment changes, so that updates in time-varying covariates are also mapped onto the coarsened scale. With coarsen = "floor", updates are mapped down to the beginning of the containing interval; with coarsen = "ceiling", they are mapped up to the end of the containing interval. After coarsening, emulated trial start times remain tied only to treatment initiation times on the coarsened scale.

The function algorithmic workflow:

1. Identifies trial start times: the earliest cohort start time plus all times at which any individual initiates treatment (tvtrt = 0 -> 1).
2. Augments each individual's trajectory by splitting intervals at these trial start times so that each at-risk subject has an explicit row starting at each trial time.
3. For each trial, includes individuals who are under observation at the trial start and have not yet started treatment, and constructs trial-specific baseline versions of treatment and covariates (with suffix .base), new follow-up times start.new, stop.new, event.
4. Implements artificial censoring by restricting to rows where the current treatment equals the baseline treatment in that trial.

Value

A seqem object with components:

- data_seqorig : stacked emulated trials **before** artificial censoring, with columns: id, trial, trial_time, rownum, visit, original start, stop, event, tvtrt, covariates, and triap-specific follow-up intervals start.new, stop.new, event, trial-specific baseline treatment and covariates {tvtrt}.base, {covs}.base for covariates in covs.

- `data_seqem` : stacked emulated trials **after** artificial censoring, obtained by restricting to rows by forcing `{tvtrt} == {tvtrt}.base`. This is the dataset typically used for downstream `seqcox()` or `coxiv_seq()` with sequential trials emulation framework.
- `summary` : a character string describing the `seqem` object, used by `print.seqem()`.
- `trial_summary` : a per-trial summary `data.frame` with one row per emulated trial and columns: `trial` (trial index), `t0` (trial start time), `n` (unique individuals), `rows` (person-time rows), `treated` (unique baseline-treated individuals), and `events` (terminal events) in that trial.

References

1. Gran JM, Roysland K, Wolbers M, Didelez V, Sterne JA, Ledergerber B, Furrer H, von Wyl V, Aalen OO. A sequential Cox approach for estimating the causal effect of treatment in the presence of time-dependent confounding applied to data from the Swiss HIV Cohort Study. *Statistics in Medicine*. 2010 Nov 20;29(26):2757-68.
2. Keogh RH, Gran JM, Seaman SR, Davies G, Vansteelandt S. Causal inference in survival analysis using longitudinal observational data: Sequential trials and marginal structural models. *Statistics in Medicine*. 2023 Jun 15;42(13):2191-2225.

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {
  data("heart", package = "survival")

  fit_seqem <- seqem(data = heart, start = "start", stop = "stop",
    event = "event", tvtrt = "transplant", id = "id",
    covs = c("age", "surgery"), coarsen = "floor", cbin_width = 7)

  print(fit_seqem)
}
```

---

vascular *vascular: longitudinal vascular surgery follow-up data*

Description

A synthetic longitudinal dataset designed to mimic repeated follow-up in a vascular surgery study. The data are stored in long start-stop format for survival analyses with a time-varying exposure and a time-varying binary prognostic factor. In particular, `reint` indicates reintervention surgery status over time and `endoleak` indicates time-varying endoleak status over time. Both are coded so that once the status becomes 1 it remains 1 at later visits.

Usage

```
data(vascular)
```

Format

A data frame with variables:

- `id` : Unique patient identifier
- `surv.time` : Observed overall follow-up time to death or censoring
- `death` : Terminal event indicator
- `time.start` : Interval start time
- `time.stop` : Interval stop time
- `event` : Counting-process event indicator for the interval
- `reint` : Time-varying reintervention status at the current visit
- `endoleak` : Time-varying endoleak status at the current visit
- `iv` : Baseline instrumental variable for reintervention propensity
- `smoke` : Baseline smoking status indicator
- `age` : Baseline age
- `diameter` : Baseline aneurysm diameter
- `gender` : Baseline gender indicator
- `hyper` : Baseline hypertension indicator
- `obese` : Baseline obesity indicator
- `reintlag1` : Lagged reintervention status from the previous interval
- `visit` : Visit number

Source

Synthetic data generated to mimic the distributional features of a vascular surgery longitudinal follow-up study.

`VitD` *VitD: cohort data on vitamin D and mortality*

Description

This dataset is mirrored from the **ivtools** package (version 2.3.0). The upstream documentation notes the data originate from a real cohort study and were modified for public availability.

Usage

```
data(VitD)
```

Format

A data frame with variables:

- `age` : Age at baseline
- `filaggrin` : Indicator of filaggrin mutation
- `vitd` : Serum 25-OH-D (nmol/L) at baseline
- `time` : Follow-up time
- `death` : Death indicator during follow-up

Source

Mirrored from **ivtools** (LGPL (>= 3)).

References

Sjolander, Arvid and Martinussen, Torben. "Instrumental Variable Estimation with the R Package ivtools" Epidemiologic Methods, vol. 8, no. 1, 2019, pp. 20180024.

Index