Package 'rpsurv'

November 6, 2026

Title Fast Royston-Parmar Flexible Parametric Survival Models

Version 0.8.2

Description Fits Royston-Parmar flexible parametric survival models (Royston and Parmar, 2002) <doi:10.1002/sim.1203> on the log cumulative hazard, odds, and probit scales, with time-varying (non-proportional) covariate effects via restricted cubic splines in log time, and smooth non-linear covariate effects via restricted cubic splines in the covariate, each with its own argument. The likelihood and its analytic gradient are evaluated in C++ with 'RcppParallel', giving large speedups over pure-R implementations for large datasets.

License MIT + file LICENSE

URL https://github.com/ielbadisy/rpsurv

BugReports https://github.com/ielbadisy/rpsurv/issues

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports Rcpp, RcppParallel, survival, stats, graphics, grDevices

LinkingTo Rcpp, RcppParallel

SystemRequirements GNU make

Suggests rstpm2, flexsurv, numDeriv, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

LazyData true

NeedsCompilation yes

Author Imad El Badisy [aut, cre]

Maintainer Imad El Badisy <elbadisyimad@gmail.com>

Repository CRAN

Date/Publication 2026-11-06 14:20:02 UTC

Contents

brcancer . . . 2
coxsnell_plot . . . 3
heart . . . 4
km_compare_plot . . . 4
nlecurve . . . 6
plot.nlecurve . . . 7
plot.rpsurv . . . 8
plot.tvecurve . . . 9
predict.rpsurv . . . 11
residuals.rpsurv . . . 12
rmst_diff . . . 13
rpsurv . . . 14
tvecurve . . . 16
veteran . . . 17

Index **18**

`brcancer` *German breast cancer data*

Description

Recurrence-free survival for 686 women in a randomized trial of hormonal therapy for breast cancer (Sauerbrei and Royston), used throughout the package's documentation and vignette as the baseline `rpsurv()` fitting example. Identical to `rstpm2::brcancer`, bundled here so examples do not require `rstpm2` to be installed.

Usage

```
brcancer
```

Format

A data frame with 686 rows and 14 variables:

id subject identifier

hormon hormonal therapy (0 = no, 1 = yes)

x1 age, years

x2 menopausal status

x3 tumour size, mm

x4 tumour grade

x5 number of positive lymph nodes

x6 progesterone receptor, fmol

x7 estrogen receptor, fmol

rectime recurrence-free survival time, days

censrec event indicator (1 = recurrence/death, 0 = censored)

x4a tumour grade >= 2

x4b tumour grade == 3

x5e exp(-0.12 * x5)

Source

https://www.stata-press.com/data/r11/brcancer.dta, as redistributed by the **rstpm2** package.

`coxsnell_plot` *Cox-Snell residual diagnostic plot*

Description

Plots the Nelson-Aalen cumulative hazard of the Cox-Snell residuals against the residuals themselves. Under a correctly specified model the residuals are a censored sample from a unit exponential distribution, so the points should lie close to the y = x reference line. `plot(fit, type = "coxsnell")` draws the same plot.

Usage

```
coxsnell_plot(object, ...)
```

Arguments

`object`	a fitted `"rpsurv"` object.
`...`	further arguments passed to `graphics::plot()`. Both axes default to the same range starting at 0, so the reference line is the diagonal of the plot.

Details

For counting-process data (`Surv(start, stop, status)`), each row's residual is `H(stop) - H(start)`. Given survival to `start`, it is unit exponential and censored at its observed value when the row has no event, so the same check applies row by row. Tied residuals are handled by the Nelson-Aalen estimator.

Value

Called for its side effect of drawing the diagnostic plot. Invisibly returns a `data.frame` with columns `time` (the sorted distinct Cox-Snell residuals) and `cumhaz` (their Nelson-Aalen cumulative hazard), the coordinates of the plotted step function.

heart *Stanford heart transplant data*

Description

Survival of patients on the waiting list for the Stanford heart transplant program (Crowley and Hu, 1977), already in counting-process (`start`, `stop`, `event`) format with a time-varying `transplant` covariate: a patient's row is split at the moment of transplantation, if it occurred. This makes it a natural, real (rather than constructed) illustration of `rpsurv()`'s support for genuine time-varying covariates via `Surv(start, stop, status)`. Identical to `survival::heart`, bundled here under the same name for convenience.

Usage

```
heart
```

Format

A data frame with 172 rows and 8 variables:

start interval start time, days

stop interval stop time, days

event event indicator at `stop` (1 = death, 0 = censored)

age age minus 48 years

year year of acceptance into the program, in years since 1967

surgery prior bypass surgery (0 = no, 1 = yes)

transplant transplant status during this interval (0 = no, 1 = yes)

id patient identifier

Source

Crowley, J. and Hu, M. (1977) Covariance analysis of heart transplant survival data. *Journal of the American Statistical Association*, 72, 27-36, as redistributed by the **survival** package.

km_compare_plot *Compare the fitted survival curve against the Kaplan-Meier estimate*

Description

A calibration / goodness-of-fit diagnostic: overlays the model-based survival curve on the nonparametric Kaplan-Meier estimate, overall or by groups of a covariate. Close agreement supports the chosen spline df and scale; systematic divergence suggests more baseline df, a different `scale`, or a `tve` term is needed. `plot(fit, type = "km")` draws the same plot.

Usage

```
km_compare_plot(
  object,
  by = NULL,
  col = NULL,
  n_grid = 200,
  legend = TRUE,
  ...
)
```

Arguments

object	a fitted "rpsurv" object.
by	optional grouping: the name of a variable of the model formula, or a vector with one value per observation used in the fit (rows with missing values dropped), for example cut(age, 3). At most 10 groups. Each group gets its Kaplan-Meier curve and its own model curve, averaged over the subjects of that group.
col	colours. Without by, two colours for the Kaplan-Meier and the model curve (default black and red); with by, one colour per group (both curves of a group share it and differ by line type).
n_grid	number of time points at which the model curve is evaluated, from 0 to the last follow-up time.
legend	TRUE (default; the legend goes in the plot corner the curves cross least), FALSE for none, or a position keyword accepted by graphics::legend().
...	further arguments passed to graphics::plot().

Details

The Kaplan-Meier curve estimates the marginal survival of the group, so the model curve is the matching population-averaged (direct-adjusted) estimand, not the survival curve at the mean covariate profile (which is a different, generally higher or lower, curve when covariates vary):

- Right-censored data: the mean over the group's subjects of their predicted survival curves, mean_i S(t | x_i). Kaplan-Meier estimates the same curve when censoring does not depend on the covariates; when it does, the two can differ slightly even for a correct model.
- Counting-process data (Surv(start, stop, status), left truncation or time-varying covariates): the Kaplan-Meier curve is then a product of conditional survival over the rows at risk, so the model curve is exp(-A(t)), where A cumulates the average model hazard of the rows at risk at each time (the model counterpart of the Nelson-Aalen estimator, computed on a fine time grid). A mean of S(t | x_i) from time 0 would not be comparable, because subjects enter the sample only after surviving to their entry time.

Value

Called for its side effect of drawing the comparison plot. Invisibly returns a data.frame with columns group, curve ("km" or "model"), time and surv, the plotted coordinates.

Examples

```
fit <- rpsurv(survival::Surv(time, status) ~ trt + celltype, data = veteran)
km_compare_plot(fit, by = "celltype")
```

`nlecurve` *Non-linear effect component of a covariate*

Description

Returns the fitted non-linear effect g(x) - g(ref) of a covariate named in `nle`, on the linear-predictor scale, with a delta-method 95% confidence interval. Because the spline enters the linear predictor additively and does not depend on time, this is an exact linear combination of the coefficients. On the hazard scale, and when `var` is not also in `tve`, it is the log hazard ratio of `x` versus `ref` at every time. When `var` is also in `tve`, it is the time-constant component of the effect, separate from the time-varying component returned by `tvecurve()`.

Usage

```
nlecurve(object, var, at = NULL, ref = NULL, se.fit = TRUE)
```

Arguments

`object`	a fitted `"rpsurv"` object.
`var`	name of a covariate in `object$nle`.
`at`	numeric values of the covariate at which to evaluate g. Defaults to 50 points over the observed range.
`ref`	reference value; `g(ref)` is subtracted. Defaults to the median of the covariate in the fitted data.
`se.fit`	logical; add `se`, `lower`, and `upper`.

Value

a data frame of class `"nlecurve"` with columns `x` and `est`, and optionally `se`, `lower`, and `upper`; see `plot.nlecurve()`.

Examples

```
set.seed(1)
n <- 500
x <- runif(n, 0, 3)
dat <- data.frame(time = rexp(n, 0.2 * exp((x - 1.5)^2 / 2)),
                  status = 1, x = x)
fit <- rpsurv(survival::Surv(time, status) ~ x, data = dat, nle = "x")
g <- nlecurve(fit, "x", ref = 1.5)
head(g)
plot(g)
```

`plot.nlecurve` *Plot a non-linear effect*

Description

Draws the output of `nlecurve()`: g(x) - g(ref) against the covariate, with its 95% confidence band and a reference line at no effect.

Usage

```
## S3 method for class 'nlecurve'
plot(
  x,
  exp = !identical(attr(x, "scale"), "normal"),
  ci = TRUE,
  log = if (exp) "y" else "",
  col = 1,
  ref.line = TRUE,
  add = FALSE,
  xlab = NULL,
  ylab = NULL,
  rug = TRUE,
  ...
)
```

Arguments

`x`	a `"nlecurve"` object returned by `nlecurve()`.
`exp`	logical; exponentiate the estimate and band. Defaults to `TRUE` on the hazard and odds scales, `FALSE` on the normal scale.
`ci`	logical; draw the confidence band when available.
`log`	passed to `graphics::plot()`; `"y"` (log y axis) when `exp` is `TRUE`, `""` otherwise.
`col`	colour of the curve and band.
`ref.line`	logical; draw a dashed line at 1 (`exp = TRUE`) or 0.
`add`	logical; add to the current plot instead of starting a new one.
`xlab, ylab`	axis labels.
`rug`	logical; mark the reference value on the x axis.
`...`	further arguments passed to `graphics::plot()`.

Value

Called for its side effect. Invisibly returns `x`.

Examples

```
fit <- rpsurv(survival::Surv(time, status) ~ trt + karno, data = veteran,
              nle = "karno")
plot(nlecurve(fit, "karno", ref = 50))
```

`plot.rpsurv` *Plot a Royston-Parmar model*

Description

Draws predicted survival, hazard or cumulative hazard curves for covariate profiles, or one of the two goodness-of-fit diagnostics: `type = "coxsnell"` calls `coxsnell_plot()` and `type = "km"` calls `km_compare_plot()`.

Usage

```
## S3 method for class 'rpsurv'
plot(
  x,
  newdata = NULL,
  type = c("survival", "hazard", "cumhaz", "coxsnell", "km"),
  times = NULL,
  ci = TRUE,
  col = NULL,
  legend = NULL,
  by = NULL,
  ...
)
```

Arguments

`x`	a fitted `"rpsurv"` object.
`newdata`	covariate profile(s) to plot, one row per curve, on the original variables of the model formula (see `predict.rpsurv()`). Defaults to a single profile with each numeric variable at its mean and each factor, character or logical variable at its most common value in the fitted data.
`type`	one of `"survival"`, `"hazard"`, `"cumhaz"` (predicted curves), `"coxsnell"` (Cox-Snell residual plot, see `coxsnell_plot()`) or `"km"` (Kaplan-Meier vs model, see `km_compare_plot()`). The diagnostics use the fitted data, so `newdata`, `times` and `ci` do not apply to them.
`times`	times at which to evaluate the curve(s). Defaults to 100 points from the first event time to the last follow-up time, the range over which the model is estimated.
`ci`	logical; add a 95% delta-method confidence band (pointwise).

col	colour(s), recycled over rows of newdata.
legend	TRUE, FALSE, or a position keyword accepted by graphics::legend() for the legend identifying the curves, labelled by the newdata columns that differ between rows. By default a legend is drawn when there are several curves, in the plot corner that the curves cross least. For type = "km" the legend is drawn by default.
by	for type = "km" only: optional grouping, the name of a variable of the model formula or a vector with one value per observation used in the fit (see km_compare_plot()).
...	further arguments passed to graphics::plot(), such as ylim or main (and n_grid for type = "km").

Value

Called for its side effect of drawing the plot. For the diagnostics, invisibly returns the value of coxsnell_plot() or km_compare_plot(). For predicted curves, invisibly returns the data.frame of predictions produced by predict.rpsurv(), with columns id (row of newdata), time, est, and, when ci = TRUE, lower and upper for the 95% confidence band. These are exactly the plotted values.

Examples

```
fit <- rpsurv(survival::Surv(time, status) ~ trt + celltype, data = veteran)
plot(fit, newdata = data.frame(trt = 1, celltype = c("squamous", "adeno")))
plot(fit, type = "coxsnell")
plot(fit, type = "km", by = "celltype")
```

---

plot.tvecurve *Plot a time-varying effect*

Description

Draws the output of tvecurve(): the effect against time, with its 95% confidence band and a reference line at no effect.

Usage

```
## S3 method for class 'tvecurve'
plot(
  x,
  exp = attr(x, "type") != "link",
  ci = TRUE,
  log = if (exp) "y" else "",
  col = 1,
  ref.line = TRUE,
  add = FALSE,
  xlab = "Time",
```

```
  ylab = NULL,
  ...
)
```

Arguments

x	a "tvecurve" object returned by tvecurve().
exp	logical; exponentiate the estimate and band. Defaults to TRUE for type = "hr" (a hazard ratio on every model scale, since tvecurve() computes it from the fitted hazards) and FALSE for type = "link".
ci	logical; draw the confidence band when available.
log	passed to graphics::plot(); "y" (log y axis) when exp is TRUE, "" otherwise.
col	colour of the curve and band.
ref.line	logical; draw a dashed line at 1 (exp = TRUE) or 0.
add	logical; add to the current plot instead of starting a new one.
xlab, ylab	axis labels.
...	further arguments passed to graphics::plot().

Details

The default y label states what is drawn. For type = "hr" it is the hazard ratio of var = ref + 1 versus var = ref (other covariates at 0). It is not a ratio "per unit" of var: with a time-varying effect the derivative of the linear predictor depends on var, so the hazard ratio changes with ref, and off the hazard scale it also depends on the other covariates. For type = "link" it is beta(t), the effect per unit of var on the model scale: a log cumulative hazard ratio (scale = "hazard"), a log cumulative odds ratio ("odds"), or a shift of the probit of the failure probability ("normal"); exponentiated, the first two are a cumulative hazard ratio and a cumulative odds ratio.

Value

Called for its side effect. Invisibly returns x.

Examples

```
fit <- rpsurv(survival::Surv(time, status) ~ trt + karno, data = veteran,
              tve = "karno")
plot(tvecurve(fit, "karno", ref = 50))
```

predict.rpsurv *Predict from a fitted Royston-Parmar model*

Description

Predict from a fitted Royston-Parmar model

Usage

```
## S3 method for class 'rpsurv'
predict(
  object,
  newdata = NULL,
  newdata0 = NULL,
  times = NULL,
  type = c("survival", "hazard", "cumhaz", "link", "hr", "sdiff"),
  se.fit = FALSE,
  ...
)
```

Arguments

object	a fitted "rpsurv" object.
newdata	a data frame of covariate values, one row per subject, on the original variables of the model formula: factors (or character values) by their levels, numeric variables untransformed, as for survival::coxph(). Factor levels not seen when fitting give an error. A data frame holding the design columns instead (for example celltypeadeno for the treatment-contrast indicator of a factor celltype) is also accepted. Defaults to the covariate values used to fit the model. For type %in% c("hr", "sdiff"), this is the *numerator*/contrast covariate profile.
newdata0	a data frame of covariate values, the *reference* profile for type %in% c("hr", "sdiff"), with the same number of rows as newdata. Required (and only used) for those two types.
times	numeric vector of times at which to predict. Defaults to a grid over the observed follow-up.
type	one of "survival", "hazard", "cumhaz", "link", "hr" or "sdiff".
	"hr" The model-implied instantaneous hazard ratio, hazard(newdata) / hazard(newdata0), computed as the ratio of two type = "hazard" predictions. This is the correct contrast under a time-varying effect (tve), where the naive exp(eta1 - eta0) is not the instantaneous hazard ratio in general.
	"sdiff" The survival difference, survival(newdata) - survival(newdata0).
se.fit	logical; if TRUE, adds lower/upper 95% confidence limits via the delta method: on the linear-predictor scale for "survival"/"cumhaz"/"link"; on the log scale for "hazard"/ "hr" (so limits stay positive); on the natural scale for "sdiff".
...	unused.

Value

a long-format data frame with columns `id`, `time`, `est`, and optionally `lower`/`upper`.

Examples

```
dat <- data.frame(
  time = rexp(200, 0.2), status = rbinom(200, 1, 0.7), x1 = rbinom(200, 1, 0.5)
)
fit <- rpsurv(survival::Surv(time, status) ~ x1, data = dat, tve = "x1")
tt <- seq(0.5, 4, length.out = 20)

hr <- predict(fit,
  newdata = data.frame(x1 = 1), newdata0 = data.frame(x1 = 0),
  times = tt, type = "hr", se.fit = TRUE
)
head(hr)

sd <- predict(fit,
  newdata = data.frame(x1 = 1), newdata0 = data.frame(x1 = 0),
  times = tt, type = "sdiff", se.fit = TRUE
)
head(sd)
```

`residuals.rpsurv` *Residuals for a Royston-Parmar model*

Description

Residuals for a Royston-Parmar model

Usage

```
## S3 method for class 'rpsurv'
residuals(object, type = c("coxsnell", "martingale", "deviance"), ...)
```

Arguments

`object`	a fitted `"rpsurv"` object.
`type`	`"coxsnell"` (Cox-Snell residuals, `r_i = H(stop_i) - H(entry_i)`), `"martingale"` (`status_i - r_i`), or `"deviance"` (the usual signed transform of the martingale residual, roughly symmetric around 0 for a well-fitting model).
`...`	unused.

Value

numeric vector of residuals, one per observation used to fit the model.

`rmst_diff` *Restricted mean survival time difference between two covariate profiles*

Description

Integrates the survival difference `sdiff = S(t | newdata) - S(t | newdata0)` from 0 to `tau` by the trapezoidal rule, with an optional delta-method standard error and 95% confidence interval.

Usage

```
rmst_diff(object, newdata, newdata0, tau, se.fit = FALSE, n_grid = 200)
```

Arguments

`object`	a fitted `"rpsurv"` object.
`newdata`	a one-row data frame, the contrast covariate profile (on the original variables, as in `predict.rpsurv()`).
`newdata0`	a one-row data frame, the reference covariate profile.
`tau`	numeric restriction time.
`se.fit`	logical; if `TRUE`, adds a delta-method standard error and 95% confidence interval. The delta method is applied directly to the trapezoidal-weighted sum of the survival-difference gradients (a linear combination of the model coefficients' asymptotic normal distribution stays normal), not by resampling.
`n_grid`	number of grid points used for the trapezoidal integration (default 200).

Value

a one-row data frame with columns `est` and, if `se.fit = TRUE`, `se`, `lower`, `upper`.

Examples

```
dat <- data.frame(
  time = rexp(200, 0.2), status = rbinom(200, 1, 0.7), x1 = rbinom(200, 1, 0.5)
)
fit <- rpsurv(survival::Surv(time, status) ~ x1, data = dat, tve = "x1")
rmst_diff(fit, newdata = data.frame(x1 = 1), newdata0 = data.frame(x1 = 0),
          tau = 4, se.fit = TRUE)
```

`rpsurv` *Fit a Royston-Parmar flexible parametric survival model*

Description

Fit a Royston-Parmar flexible parametric survival model

Usage

```
rpsurv(
  formula,
  data,
  df = 4,
  knots = NULL,
  tve = NULL,
  tve.df = 3,
  nle = NULL,
  nle.df = 3,
  scale = c("hazard", "odds", "normal"),
  control = list()
)
```

Arguments

`formula`	a survival formula. Either `Surv(time, status) ~ covariates` for standard right-censored data, or `Surv(start, stop, status) ~ covariates` (counting-process form) for left-truncated data and/or genuine **time-varying covariates** (a covariate whose *value* changes over follow-up): split each subject's follow-up into intervals of constant covariate values and pass one row per interval. See Details.
`data`	a data frame.
`df`	degrees of freedom for the baseline spline in log time (number of interior knots is `df - 1`). Default 4.
`knots`	optional numeric vector of ALL baseline knots (boundary knots first/last), overriding `df`.
`tve`	character vector of covariate names allowed a **time-varying effect** (non-proportional hazards/odds/probit): the covariate's *coefficient* beta(t) is modelled as its own spline in log time, while its value is still fixed for a subject. This is distinct from a time-varying *covariate* (see `formula`); the two can be combined.
`tve.df`	degrees of freedom for each time-varying effect spline: a single value for all `tve` covariates, or a vector named by covariate. Default 3.
`nle`	character vector of numeric covariate names allowed a smooth **non-linear effect** g(x), modelled as a restricted cubic spline in the covariate's value (linear beyond the boundary knots, placed at the minimum and maximum of the covariate; interior knots at equally spaced centiles). The effect does not change over

time. A covariate can be in both `nle` and `tve`: its log-hazard contribution is then g(x) + beta(t) x, and `nlecurve()` and `tvecurve()` return the two components separately.

`nle.df` degrees of freedom for each non-linear effect spline (1 is linear): a single value for all `nle` covariates, or a vector named by covariate. Default 3.

`scale` one of `"hazard"` (proportional hazards, the Royston-Parmar default), `"odds"` (proportional odds) or `"normal"` (probit).

`control` list of control parameters passed to `stats::optim()`. The optimisation runs on an internal orthogonalised parameterisation (see Details), so parameter-scale settings such as `parscale` or `ndeps` refer to that parameterisation.

Value

an object of class `"rpsurv"`.

Time-varying effect vs. time-varying covariate

These are two distinct extensions and `rpsurv()` supports both, separately or combined:

- **Time-varying effect** (non-proportional hazards): a covariate's value is fixed for a subject but its *association with the outcome* changes over time, e.g. a treatment effect that fades. Request this with `tve`.
- **Time-varying covariate**: a covariate's *value* itself changes during follow-up, e.g. a lab measurement updated at clinic visits. Represent this in the data as counting-process (start, stop] intervals, one row per interval with the covariate value held constant within it, and fit with `Surv(start, stop, status) ~ ...`. `rpsurv()` handles the corresponding left-truncated likelihood automatically.

Non-linear effect vs. time-varying effect

`tve` lets a covariate's effect change with *time*; `nle` lets it change with the covariate's own *value*. Each has its own argument and spline degrees of freedom, so the two components are fitted, reported, and extracted separately: `tvecurve()` returns beta(t) and `nlecurve()` returns g(x) - g(ref).

Estimation

The log-likelihood and its analytic gradient are maximised with BFGS over a linear reparameterisation of the coefficients in which the design matrix has orthogonal, unit-scale columns (from its QR decomposition). This keeps the problem well conditioned when covariates on large scales multiply the time spline basis in `tve` terms. The Hessian is obtained by central differences of the analytic gradient in that parameterisation, and the coefficients and their variance-covariance matrix are mapped back exactly to the scale of the original design.

tvecurve *Time-varying effect component of a covariate*

Description

Returns the fitted time-varying effect of a covariate named in `tve`, with a delta-method 95% confidence interval.

Usage

```
tvecurve(
  object,
  var,
  times = NULL,
  type = c("hr", "link"),
  ref = 0,
  se.fit = TRUE
)
```

Arguments

`object`	a fitted `"rpsurv"` object.
`var`	name of a covariate in `object$tve`.
`times`	times at which to evaluate the effect. Defaults to 50 points over the observed event times.
`type`	`"hr"` or `"link"`.
`ref`	reference value of `var` for `type = "hr"`. Default 0.
`se.fit`	logical; add `se`, `lower`, and `upper`.

Details

- `type = "hr"` (default): the log of the instantaneous hazard ratio at each time for `var = ref + 1` versus `var = ref`, as in `predict.rpsurv()` with `type = "hr"`. Other covariates are set to 0; on the hazard scale they cancel unless they also have a time-varying effect. If `var` is also in `nle`, its non-linear component is held fixed, so the result is the time-varying component only (see `nlecurve()` for the other one).
- `type = "link"`: the coefficient function beta(t) on the linear-predictor (log cumulative hazard, log cumulative odds, or probit) scale, the main effect plus its spline in log time; when `var` is also in `nle` the main effect belongs to g, so beta(t) is the spline part only.

Value

a data frame of class `"tvecurve"` with columns `time` and `est`, and optionally `se`, `lower`, and `upper`; see `plot.tvecurve()`.

Examples

```
fit <- rpsurv(survival::Surv(time, status) ~ trt + karno, data = veteran,
              tve = "karno", nle = "karno")
tvecurve(fit, "karno", times = c(30, 90, 180), ref = 50)
plot(tvecurve(fit, "karno", ref = 50))
```

---

veteran *Veterans' Administration lung cancer trial*

Description

A randomized trial comparing two treatment regimens for lung cancer (Kalbfleisch and Prentice). `celltype` is a well-known example of a non-proportional-hazards effect: its hazard ratio changes materially over follow-up, which makes this dataset a natural illustration of `rpsurv()`'s `tve` (time-varying effect) argument. Identical to `survival::veteran`, bundled here under the same name for convenience.

Usage

```
veteran
```

Format

A data frame with 137 rows and 8 variables:

trt treatment (1 = standard, 2 = test)

celltype tumour cell type: squamous, smallcell, adeno, large

time survival time, days

status event indicator (1 = dead, 0 = censored)

karno Karnofsky performance score (0-100)

diagtime months from diagnosis to randomization

age age, years

prior prior therapy (0 = no, 10 = yes)

Source

Kalbfleisch, J. and Prentice, R. (1980) *The Statistical Analysis of Failure Time Data*. Wiley, New York, as redistributed by the **survival** package.

Index

∗ **datasets**
brcancer, 2
heart, 4
veteran, 17

brcancer, 2

coxsnell_plot, 3
coxsnell_plot(), *8, 9*

graphics::legend(), *5*, *9*
graphics::plot(), *3*, *5*, *7*, *9, 10*

heart, 4

km_compare_plot, 4
km_compare_plot(), *8, 9*

nlecurve, 6
nlecurve(), *7*, *15, 16*

plot.nlecurve, 7
plot.nlecurve(), *6*
plot.rpsurv, 8
plot.tvecurve, 9
plot.tvecurve(), *16*
predict.rpsurv, 11
predict.rpsurv(), *8, 9*, *13*, *16*

residuals.rpsurv, 12
rmst_diff, 13
rpsurv, 14

stats::optim(), *15*
survival::coxph(), *11*

tvecurve, 16
tvecurve(), *6*, *9, 10*, *15*

veteran, 17