Package 'rpsurv'

September 11, 2026

Title Fast Royston-Parmar Flexible Parametric Survival Models

Version 0.7.1

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. 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 2026-09-11 13:50:02 UTC

Contents

`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. A well-fitting model gives points scattered around the y = x line.

Usage

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

Arguments

`object`	a fitted `"rpsurv"` object.
`...`	further arguments passed to `graphics::plot()`.

Value

Called for its side effect of drawing the diagnostic plot. Invisibly returns a `data.frame` with columns `time` (the sorted 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-predicted survival curve on the non-parametric Kaplan-Meier estimate, optionally by strata of a categorical 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.

Usage

```
km_compare_plot(object, by = NULL, col = c("black", "red"), ...)
```

Arguments

`object`	a fitted `"rpsurv"` object.
`by`	optional name of a covariate in the fitted data to stratify by (each level gets its own KM curve and predicted curve, at that level's mean of the other covariates).
`col`	colours for KM (solid step) vs model (dashed) curves.
`...`	further arguments passed to `graphics::plot()`.

Value

No return value, called for its side effect of drawing the Kaplan-Meier versus fitted-survival comparison plot.

`plot.rpsurv` *Plot predicted curves from a Royston-Parmar model*

Description

Plot predicted curves from a Royston-Parmar model

Usage

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

Arguments

`x`	a fitted `"rpsurv"` object.
`newdata`	covariate profile(s) to plot, one row per curve. Defaults to a single profile at the mean of each covariate.
`type`	one of `"survival"`, `"hazard"`, `"cumhaz"`.
`times`	times at which to evaluate the curve(s).
`ci`	logical; add a 95% confidence band (ignored for `type = "hazard"`).
`col`	colour(s), recycled over rows of `newdata`.
`...`	further arguments passed to `graphics::plot()`.

Value

Called for its side effect of drawing the predicted curve(s). Invisibly returns the `data.frame` of predictions produced by `predict.rpsurv()`, with columns `id` (row of `newdata`), `time`, `est`, and, when `ci = TRUE` and `type != "hazard"`, `lower` and `upper` for the 95% confidence band.

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. 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.
`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,
  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. 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()`.

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.

`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