

Package ‘survParamSim’

May 9, 2026

Type Package

Title Parametric Survival Simulation with Parameter Uncertainty

Version 0.1.7

Description Perform survival simulation with parametric survival model generated from 'survreg' function in 'survival' package.

In each simulation coefficients are resampled from variance-covariance matrix of parameter estimates to capture uncertainty in model parameters.

Prediction intervals of Kaplan-Meier estimates and hazard ratio of treatment effect can be further calculated using simulated survival data.

License GPL-3

Encoding UTF-8

Depends R (>= 3.4.0),

Imports broom, dplyr, eha, forcats, ggplot2, lifecycle (>= 0.2.0), magrittr (>= 1.5), methods, mvtnorm, purrr, rlang, survival (>= 2.43), tibble, tidyr (>= 1.1),

Suggests knitr, rmarkdown, survminer, testthat (>= 3.1.0), vdiff, withr, flexsurv (>= 2.2),

RoxygenNote 7.2.3

Language en-US

URL <https://github.com/yoshidk6/survParamSim>,
<https://yoshidk6.github.io/survParamSim/>

BugReports <https://github.com/yoshidk6/survParamSim/issues>

VignetteBuilder knitr

RdMacros lifecycle

Config/testthat/edition 3

NeedsCompilation no

Author Kenta Yoshida [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-4967-3831>>),
Laurent Claret [aut]

Maintainer Kenta Yoshida <yoshida.kenta.6@gmail.com>

Repository CRAN

Date/Publication 2025-05-02 19:30:02 UTC

Contents

calc_ave_hr_pi	2
calc_ave_km_pi	3
extractpi	5
extractrawsim	6
extract_median_surv	7
plot_hr_pi	7
plot_km_pi	8
print.survparamsim.hrpi	8
survparamsim	9
surv_param_sim_pre_resampled	11

Index 13

calc_ave_hr_pi	<i>Generate hazard ratio with prediction intervals from parametric bootstrap simulation</i>
----------------	---

Description

Generate hazard ratio with prediction intervals from parametric bootstrap simulation

Usage

```
calc_ave_hr_pi(
  sim,
  trt,
  group = NULL,
  pi.range = 0.95,
  calc.obs = TRUE,
  trt.assign = c("default", "reverse"),
  simtimelast = NULL,
  boot.subj = TRUE
)
```

```
calc_hr_pi(
  sim,
  trt,
  group = NULL,
  pi.range = 0.95,
  calc.obs = TRUE,
  trt.assign = c("default", "reverse")
)
```

Arguments

sim	A survparamsim class object generated by <code>surv_param_sim()</code> function.
trt	A string to specify which column define treatment status to calculate HR.
group	Optional string(s) to specify grouping variable(s). You will have faceted histograms for these variables in <code>plot_hr_pi()</code> function.
pi.range	Prediction interval for simulated HR.
calc.obs	A logical to specify whether to calculate HR for the observed data. Need be set as FALSE if survival information in the newdata is dummy.
trt.assign	Specify which of the categories of trt need to be considered as control group. See details below if you have more than two categories.
sintimelast	An optional numeric to specify duration for average HR calculation. If NULL (default), the last observation time in the newdata will be used.
boot.subj	Boolean to specify whether bootstrapping of subjects are performed before calculating HR. Default TRUE.

Details

`calc_hr_pi()` calculate hazard ratio using the simulated survival times with Cox proportional hazard model, while `calc_ave_hr_pi()` calculate "average" hazard ratio using the mean survival & probability density function per treatment groups.

If your trt has more than two categories/levels and want to specify which one to use as a reference group, you can convert the column into a factor in the newdata input for `surv_param_sim()`. The first level will be used as a reference group.

calc_ave_km_pi	<i>Generate Kaplan-Meier curves with prediction intervals using simulated survival time</i>
----------------	---

Description

Generate Kaplan-Meier curves with prediction intervals using simulated survival time

Usage

```
calc_ave_km_pi(
  sim,
  trt = NULL,
  group = NULL,
  pi.range = 0.95,
  calc.obs = TRUE,
  sintimelast = NULL,
  trt.assign = c("default", "reverse"),
  boot.subj = TRUE,
  calc.median.surv = FALSE
```

```

)

calc_km_pi(
  sim,
  trt = NULL,
  group = NULL,
  pi.range = 0.95,
  calc.obs = TRUE,
  simtimelast = NULL,
  trt.assign = c("default", "reverse")
)

```

Arguments

sim	A survparamsim class object generated by surv_param_sim() function.
trt	An optional string to specify which column define treatment status. You will have survival curves with different colors in plot_km_pi() function.
group	Optional string(s) to specify grouping variable(s). You will have faceted survival curves for these variables in plot_km_pi() function.
pi.range	Prediction interval for simulated survival curves.
calc.obs	A logical to specify whether KM estimates will be performed for the observed data. Need be set as FALSE if survival information in the newdata is dummy.
simtimelast	An optional numeric to specify last simulation time for survival curve. If NULL (default), the last observation time in the newdata will be used.
trt.assign	Specify which of the categories of trt need to be considered as control group. See details below if you have more than two categories. Only applicable if you will use extract_medsurv_delta_pi() to extract delta of median survival times.
boot.subj	Boolean to specify whether bootstrapping of subjects are performed before calculating HR. Default TRUE.
calc.median.surv	Whether to calculate median survival time for calc_ave_km_pi() . Default FALSE as the calculation can be long. Currently median survival calculation not implemented yet.

Details

[calc_km_pi\(\)](#) calculate survival profile using the simulated survival times with Kaplan-Meier estimates, while [calc_ave_km_pi\(\)](#) calculate "average" survival using the mean survival function per treatment groups. [calc_ave_km_pi\(\)](#) actually does not rely on Kaplan-Meier estimates as it directly uses the underlying parametric survival model, however the function has km for naming consistency.

If your trt has more than two categories/levels and want to specify which one to use as a reference group, you can convert the column into a factor in the newdata input for [surv_param_sim\(\)](#). The first level will be used as a reference group.

extractpi	<i>Functions to extract prediction intervals and observed data</i>
-----------	--

Description

Functions to extract prediction intervals and observed data

Usage

```
extract_hr_pi(hr.pi, outtype = c("long", "wide"))  
extract_km_pi(km.pi, trunc.sim.censor = TRUE)  
extract_medsurv_pi(km.pi, outtype = c("long", "wide"))  
extract_medsurv_delta_pi(km.pi, outtype = c("long", "wide"))
```

Arguments

hr.pi	a return object from calc_hr_pi() function.
outtype	Specifies whether output will be in long or wide format.
km.pi	A return object from calc_km_pi() function.
trunc.sim.censor	A logical specifying whether to truncate the simulated curve at the last time of <code>sensor.dur</code> specified in surv_param_sim() .

Details

[extract_hr_pi\(\)](#) extracts prediction intervals of simulated hazard ratios and the corresponding observed values.

[extract_km_pi\(\)](#) extracts prediction intervals of simulated Kaplan-Meier curves.

[extract_medsurv_pi\(\)](#) extracts prediction intervals of median survival times and the corresponding observed values.

[extract_medsurv_delta_pi\(\)](#) extracts prediction intervals of delta of median survival times between treatment groups

`extractrawsim`*Functions to extract raw simulated samples*

Description

Functions to extract raw simulated samples

Usage

```
extract_sim(sim)
```

```
extract_hr(hr.pi)
```

```
extract_km_obs(km.pi)
```

```
extract_medsurv(km.pi)
```

```
extract_medsurv_delta(km.pi)
```

Arguments

<code>sim</code>	A <code>survparamsim</code> class object generated by <code>surv_param_sim()</code> function.
<code>hr.pi</code>	a return object from <code>calc_hr_pi()</code> function.
<code>km.pi</code>	A return object from <code>calc_km_pi()</code> function.

Details

`extract_sim()` extracts raw survival time & event status for all simulated subjects.

`extract_hr()` extracts simulated HR for all repeated simulations. If HR was calculated based on simulated survival times with `calc_hr_pi()` function, it also returns p values for Cox regression fits, one for each group based on Wald test and another for the overall significance of the coefficient based on logrank test. The latter has the same values across treatment groups when >2 levels in treatment

`extract_km_obs()` extracts observed Kaplan-Meier curves.

`extract_medsurv()` extracts simulated median survival times for all repeated simulations

`extract_medsurv_delta()` extracts delta of median survival times between treatment groups

extract_median_surv *Functions to extract prediction intervals and observed data*

Description

[Deprecated]

Usage

```
extract_median_surv(km.pi, outtype = c("long", "wide"))
```

Arguments

km.pi A return object from [calc_km_pi\(\)](#) function.
outtype Specifies whether output will be in long or wide format.

Details

[extract_median_surv\(\)](#) was renamed to [extract_medsurv_pi\(\)](#) for function name consistency.
[extract_median_surv\(\)](#) extracts prediction intervals of median survival times and and the corresponding observed values.

plot_hr_pi *Plot simulated HR histogram(s) overlaid with prediction intervals*

Description

Plot simulated HR histogram(s) overlaid with prediction intervals

Usage

```
plot_hr_pi(hr.pi, show.obs = TRUE)
```

Arguments

hr.pi a return object from [calc_hr_pi](#) function.
show.obs A logical specifying whether to show observed HR on the plot. This will have no effect if `calc.obs` was set to FALSE in [calc_hr_pi](#).

plot_km_pi	<i>Plot Kaplan-Meier curves with prediction intervals from parametric bootstrap simulation</i>
------------	--

Description

Need to think about how to apply this for subgroups

Usage

```
plot_km_pi(km.pi, show.obs = TRUE, trunc.sim.censor = TRUE)
```

Arguments

km.pi	an output from <code>calc_km_pi</code> function.
show.obs	A logical specifying whether to show observed K-M curve on the plot. This will have no effect if <code>calc.obs</code> was set to <code>FALSE</code> in <code>calc_km_pi</code> .
trunc.sim.censor	A logical specifying whether to truncate the simulated curve at the last time of <code>sensor.dur</code> specified in <code>surv_param_sim</code> .

print.survparamsim.hrpi	<i>Methods for S3 objects in the package</i>
-------------------------	--

Description

Methods for S3 objects in the package

Usage

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

## S3 method for class 'survparamsim.hrpi'
summary(object, ...)

## S3 method for class 'survparamsim.kmpi'
print(x, ...)

## S3 method for class 'survparamsim.kmpi'
summary(object, ...)

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

Arguments

x	An object of the corresponding class
...	Additional arguments passed to methods.
object	An object of the corresponding class

survparamsim

Simulation of parametric survival model

Description

The main function(s) to generate predicted survival using a model object generated with `survival::survreg()` function.

Usage

```
surv_param_sim(
  object,
  newdata,
  n.rep = 1000,
  censor.dur = NULL,
  coef.var = TRUE,
  na.warning = TRUE
)
```

```
surv_param_sim_resample(
  object,
  newdata,
  n.rep = 1000,
  censor.dur = NULL,
  n.resample,
  strat.resample = NULL,
  coef.var = TRUE,
  na.warning = TRUE
)
```

Arguments

object	A <code>survreg</code> class object. Currently accept exponential, lognormal, weibull, loglogistic, and gaussian distributions.
newdata	A required data frame for simulation that contain covariates in the survival model. It is required even if this is the same as the one used for <code>survival::survreg</code> function. It also has to contain columns for survival information. These can be used in <code>plot_km_pi()</code> and <code>plot_hr_pi()</code> function as observed data. Survival information can be dummy data, but time need to be long enough so that simulated KM plot will be long enough for <code>plot_km_pi()</code> to draw simulated survival curves.

	Subjects with NA for covariates in survreg model will be removed from the simulation and subsequent plotting of observed data.
n.rep	An integer defining numbers of parametric bootstrap runs
sensor.dur	A two elements vector specifying duration of events censoring. Censoring time will be calculated with uniform distribution between two numbers. No censoring will be applied if NULL is provided.
coef.var	Boolean specifying whether parametric bootstrap are performed on survival model coefficients, based on variance-covariance matrix. If FALSE, prediction interval only reflects inherent variability from survival events.
na.warning	Boolean specifying whether warning will be shown if newdata contain subjects with missing model variables.
n.resample	Number of subjects for resampled simulations. If strat.resample is provided, this needs to be a vector of the length equal to the number of categories in the stratification variable.
strat.resample	String specifying stratification variable for resampling. Currently only one variable is allowed. If you need more than one, create a new variable e.g. by <code>base::interaction()</code>

Details

`surv_param_sim()` returns simulation using the provided subject in newdata as it is, while `surv_param_sim_resample` perform simulation based on resampled subjects from the dataset. The latter allows more flexibility in terms of simulating future trials with different number of subjects. Note that with `surv_param_sim_resample()`, there is no automatic safeguard to ensure certain number of subjects in each subgroup or treatment groups, which may result in inconsistent number of subjects per simulation or leads to Cox regression instability due to small N. Consider using stratified resampling in this case.

Currently we have not tested whether this function work for a survreg model with stratification variables.

Value

A `survparamsim` object that contains the original `survreg` class object, `newdata`, and a data frame for predicted survival profiles with the following columns:

- **time**: predicted event or censor time
- **event**: event status, 0=censored, 1=event
- **rep**: ID for parametric bootstrap runs
- **subj**: ID for subjects in newdata (currently original ID is not retained and subj is sequentially assigned as 1:nrow(newdata))

Examples

```
library(survival)

fit.lung <- survreg(Surv(time, status) ~ sex + ph.ecog, data = lung)

object <- fit.lung
```

```

n.rep <- 30
newdata <-
  tibble::as_tibble(dplyr::select(lung, time, status, sex, ph.ecog)) %>%
  tidyr::drop_na()
censor.dur <- c(200, 1100)

sim <- surv_param_sim(object, newdata, n.rep, censor.dur)

```

surv_param_sim_pre_resampled

Simulation of parametric survival model with an already-resampled dataset

Description

Simulation of parametric survival model with an already-resampled dataset

Usage

```

surv_param_sim_pre_resampled(
  object,
  newdata.resampled,
  newdata.orig = NULL,
  censor.dur = NULL,
  coef.var = TRUE,
  na.warning = TRUE
)

```

Arguments

object	A survreg class object. Currently accept exponential, lognormal, weibull, loglogistic, and gaussian distributions.
newdata.resampled	A required input, the already resampled dataset for simulation. This dataset must have: (a) rep variable indicating the #simulation groups, and (b) the same number of subjects per each rep
newdata.orig	An optional input needed for calculating KM and HR for the observed data.
censor.dur	A two elements vector specifying duration of events censoring. Censoring time will be calculated with uniform distribution between two numbers. No censoring will be applied if NULL is provided.
coef.var	Boolean specifying whether parametric bootstrap are performed on survival model coefficients, based on variance-covariance matrix. If FALSE, prediction interval only reflects inherent variability from survival events.
na.warning	Boolean specifying whether warning will be shown if newdata contain subjects with missing model variables.

Details

See [surv_param_sim\(\)](#) for additional details.

Value

A `survparamsim` object that contains the original `survreg` class object, `newdata`, and a data frame for predicted survival profiles.

Index

`base::interaction()`, [10](#)

`calc_ave_hr_pi`, [2](#)
`calc_ave_hr_pi()`, [3](#)
`calc_ave_km_pi`, [3](#)
`calc_ave_km_pi()`, [4](#)
`calc_hr_pi`, [7](#)
`calc_hr_pi (calc_ave_hr_pi)`, [2](#)
`calc_hr_pi()`, [3, 5, 6](#)
`calc_km_pi`, [8](#)
`calc_km_pi (calc_ave_km_pi)`, [3](#)
`calc_km_pi()`, [4–7](#)
`calculate_hr_pi (calc_ave_hr_pi)`, [2](#)
`calculate_km_pi (calc_ave_km_pi)`, [3](#)

`extract_hr (extractrawsim)`, [6](#)
`extract_hr()`, [6](#)
`extract_hr_pi (extractpi)`, [5](#)
`extract_hr_pi()`, [5](#)
`extract_km_obs (extractrawsim)`, [6](#)
`extract_km_obs()`, [6](#)
`extract_km_pi (extractpi)`, [5](#)
`extract_km_pi()`, [5](#)
`extract_median_surv`, [7](#)
`extract_median_surv()`, [7](#)
`extract_medsurv (extractrawsim)`, [6](#)
`extract_medsurv()`, [6](#)
`extract_medsurv_delta (extractrawsim)`, [6](#)
`extract_medsurv_delta()`, [6](#)
`extract_medsurv_delta_pi (extractpi)`, [5](#)
`extract_medsurv_delta_pi()`, [4, 5](#)
`extract_medsurv_pi (extractpi)`, [5](#)
`extract_medsurv_pi()`, [5, 7](#)
`extract_sim (extractrawsim)`, [6](#)
`extract_sim()`, [6](#)
`extractpi`, [5](#)
`extractrawsim`, [6](#)

`plot_hr_pi`, [7](#)
`plot_hr_pi()`, [3, 9](#)

`plot_km_pi`, [8](#)
`plot_km_pi()`, [4, 9](#)
`print.survparamsim`
 (`print.survparamsim.hrpi`), [8](#)
`print.survparamsim.hrpi`, [8](#)

`summary.survparamsim.hrpi`
 (`print.survparamsim.hrpi`), [8](#)
`summary.survparamsim.kmpi`
 (`print.survparamsim.hrpi`), [8](#)
`surv_param_sim`, [8](#)
`surv_param_sim (survparamsim)`, [9](#)
`surv_param_sim()`, [3–6, 10, 12](#)
`surv_param_sim_pre_resampled`, [11](#)
`surv_param_sim_resample`, [10](#)
`surv_param_sim_resample (survparamsim)`,
 [9](#)
`surv_param_sim_resample()`, [10](#)
`survival::survreg`, [9](#)
`survival::survreg()`, [9](#)
`survparamsim`, [9](#)
`survparamsim-methods`
 (`print.survparamsim.hrpi`), [8](#)