Package 'survParamSim'

June 2, 2022

Type Package

Title Parametric Survival Simulation with Parameter Uncertainty

Version 0.1.6

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,
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 (¿= 2.1.0),
vdiffr,
withr

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Language en-US

URL https://github.com/yoshidk6/survParamSim

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

VignetteBuilder knitr

RdMacros lifecycle

R topics documented:

`calc_hr_pi` *Generate hazard ratio with prediction intervals from parametric bootstrap simulation*

Description

Generate hazard ratio with prediction intervals from parametric bootstrap simulation

Usage

```
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 `surv_param_sim()` 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 `plot_hr_pi()` 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.

Details

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_km_pi` *Generate Kaplan-Meier curves with prediction intervals from parametric bootstrap simulation*

Description

Generate Kaplan-Meier curves with prediction intervals from parametric bootstrap simulation

Usage

```
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.

Details

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 *Functions to extract prediction intervals and observed data*

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 censor.dur 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 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

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

Details

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

extract_hr() extracts simulated HRs for all repeated simulations. 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) overlayed with prediction intervals*

Description

Plot simulated HR histogram(s) overlayed 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` *Plot Kaplan-Meier curves with prediction intervals from parametric bootstrap simulation*

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 `calc_km_pi` function.
`show.obs`	A logical specifying whether to show observed K-M curve on the plot. This will have no effect if `calc.obs` was set to `FALSE` in `calc_km_pi`.
`trunc.sim.censor`	A logical specifying whether to truncate the simulated curve at the last time of `censor.dur` specified in `surv_param_sim`.

print.survparamsim.hrpi *Methods for S3 objects in the package*

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 `survreg` 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 survival::survreg function. It also has to contain columns for survival information. These can be used in `plot_km_pi()` and `plot_hr_pi()` 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 `plot_km_pi()` 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
`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.
`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 `base::interaction()`

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