Package ‘eventPred’

June 5, 2026

Title Event Prediction

Version 0.3.0

Description Predicts enrollment and events at the design or analysis stage using specified enrollment and time-to-event models through simulations.

License GPL (>= 2)

Encoding UTF-8

Imports data.table (>= 1.14.10), magrittr (>= 2.0.0), ggplot2 (>= 3.5.1), plotly (>= 4.10.1), survival (>= 2.41-3), splines (>= 3.5.0), Matrix (>= 1.2-14), mvtnorm (>= 1.1-3), rstpm2 (>= 1.6.1), numDeriv (>= 2016.8-1.1), purrr (>= 1.0.2), flexsurv (>= 2.2.2), erify (>= 0.4.0), stats (>= 3.5.0), shiny (>= 1.7.1), rlang (>= 1.0.6), lrstat (>= 0.2.12)

Suggests knitr (>= 1.45), rmarkdown (>= 2.25), testthat (>= 3.0.0), pkgdown (>= 2.2.0)

Depends R (>= 3.5.0)

VignetteBuilder knitr

LazyData true

URL `https://kaifenglu.github.io/eventPred/`,
`https://github.com/kaifenglu/eventPred`

BugReports `https://github.com/kaifenglu/eventPred/issues`

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

NeedsCompilation no

Author Kaifeng Lu [aut, cre] (ORCID: <`https://orcid.org/0000-0002-6160-7119`>)

Maintainer Kaifeng Lu <`kaifenglu@gmail.com`>

Repository CRAN

Date/Publication 2026-06-05 07:30:08 UTC

Contents

`eventPred-package` *Event Prediction*

Description

Predicts enrollment and events at the design stage using assumed enrollment and treatment-specific time-to-event models, or at the analysis stage using blinded or unblinded data and specified enrollment and time-to-event models through simulations.

Details

Accurately predicting the date at which a target number of subjects or events will be achieved is critical for the planning, monitoring, and execution of clinical trials. The `eventPred` package provides enrollment and event prediction capabilities using assumed enrollment and treatment-specific time-to-event models at the design stage, using blinded or unblinded data and specified enrollment and time-to-event models at the analysis stage.

At the design stage, enrollment is often specified using a piecewise Poisson process with a constant enrollment rate during each specified time interval. At the analysis stage, before enrollment completion, the `eventPred` package considers several models, including the homogeneous Poisson model, the time-decay model with an enrollment rate function $\lambda(t) = (\mu/\delta)(1 - \exp(-\delta t))$, the B-spline model with the daily enrollment rate $\lambda(t) = \exp(B(t)'\theta)$, and the piecewise Poisson model. If prior information exists on the model parameters, it can be combined with the likelihood to yield the posterior distribution.

The `eventPred` package also offers several time-to-event models, including exponential, Weibull, log-logistic, log-normal, piecewise exponential, model averaging of Weibull and log-normal, spline, and cox. For time to dropout, the same set of model options are considered. If enrollment is complete, ongoing subjects who have not had the event of interest or dropped out of the study before the data cut contribute additional events in the future. Their event times are generated from the conditional distribution given that they have survived at the data cut. For new subjects that need to be enrolled, their enrollment time and event time can be generated from the specified enrollment and time-to-event models with parameters drawn from the posterior distribution. Time-to-dropout can be generated in a similar fashion.

The `eventPred` package displays the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC) and a fitted curve overlaid with observed data to help users select the most appropriate model for enrollment and event prediction. Prediction intervals in the prediction plot can be used to measure prediction uncertainty, and the simulated enrollment and event data can be used for further data exploration.

The most useful function in the `eventPred` package is `getPrediction`, which combines model fitting, data simulation, and a summary of simulation results. Other functions perform individual tasks and can be used to select an appropriate prediction model.

The `eventPred` package implements a model parameterization that enhances the asymptotic normality of parameter estimates. Specifically, the package utilizes the following parameterization to achieve this goal:

- Enrollment models
  - Poisson: $\theta = \log(\lambda)$.
  - Time-decay: $\theta = (\log(\mu), \log(\delta))'$.
  - B-spline: $\log(\lambda(t)) = B(t)'\theta$, $B(t) = (B_1(t), \ldots, B_{k+4}(t))'$ are the B-spline basis with k inner knots.
  - Piecewise Poisson: $\theta_j = \log(\lambda_j)$ for the jth time interval. The left endpoints of time intervals, denoted as `accrualTime`, are considered fixed.
- Event or dropout models

  Let x denote the covariates for a subject. Let β denote the regression coefficients and σ denote the scale parameter of the AFT model,

  $$\log(T) = \beta' x + \sigma\epsilon.$$

  - Exponential: $\log(\lambda) = \theta' x$. In other words, $\theta = -\beta$.
  - Weibull: $\log(\texttt{weibull scale}) = \theta_1' x$, $\log(\texttt{weibull shape}) = -\theta_2$. In other words, $\theta = (\beta', \log(\sigma))'$.
  - Log-logistic: For the logistic distribution of $\log(T)$, $\texttt{location} = \theta_1' x$, $\log(\texttt{scale}) = \theta_2$. In other words, $\theta = (\beta', \log(\sigma))'$.
  - Log-normal: For the normal distribution of $\log(T)$, $\texttt{mean} = \theta_1' x$, $\log(\texttt{sd}) = \theta_2$. In other words, $\theta = (\beta', \log(\sigma))'$.
  - Piecewise exponential: $\log(\lambda_j) = \theta_{1j} + \theta_2' x$ for the jth time interval, $\theta = (\theta_1', \theta_2')'$. The left endpoints of time intervals, denoted as `piecewiseSurvivalTime` for event model and `piecewiseDropoutTime` for dropout model, are considered fixed.
  - Model averaging: $\theta = (\theta_{\texttt{weibull}}', \theta_{\texttt{lnorm}}')'$. The covariance matrix for θ is structured as a block diagonal matrix, with the upper-left block corresponding to the Weibull component and the lower-right block corresponding to the log-normal component. In other words, the covariance matrix is partitioned into two distinct blocks, with no off-diagonal elements connecting the two components. The weight assigned to the Weibull component, denoted as w_1, is considered fixed.
  - Spline: Let $S(t|x)$ denote the survival function given covariates x. We model a transformation of the survival function as a cubic spine:

    $$g(S(t|x)) = c(u) + \theta_2' x,$$

where

$$c(u) = \gamma_1 + \gamma_2 u + \gamma_3 v_1(u) + \cdots + \gamma_{k+2} v_k(u)$$

is the cubic spline in $u = \log(t)$, $\theta = (\theta_1', \theta_2')'$, $\theta_1 = (\gamma_1, \ldots, \gamma_{k+2})'$, assuming k inner knots ($k =$ `knots`), and $v_1(u), \ldots, v_k(u)$ are the basis of the Royston/Parmar spline. The transformation is given as follows:

- For `scale = "hazard"`, $g(S(t)) = \log(-\log(S(t)))$.
- For `scale = "odds"`, $g(S(t)) = \log(1/S(t) - 1)$.
- For `scale = "normal"`, $g(S(t)) = -\Phi^{-1}(S(t))$.

The hazard, odds, and normal scales correspond to extensions of the Weibull, log-logistic, and log-normal distributions, respectively.

- Cox: Let $t_1 < \cdots < t_M$ denote the distinct observed event times, λ_j denote the estimated baseline hazard rate in the jth time interval, $(t_{j-1}, t_j]$, and β denote the regression coefficients (log hazard ratios) from the Cox model. The model parameters including the baseline hazards are $\theta = (\log(\lambda_1), \ldots, \log(\lambda_M), \beta^T)^T$.

The `eventPred` package uses days as its primary time unit. If you need to convert enrollment or event rates per month to rates per day, simply divide by 30.4375.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Emilia Bagiella and Daniel F. Heitjan. Predicting analysis times in randomized clinical trials. Stat in Med. 2001; 20:2055-2063.

Gui-shuang Ying and Daniel F. Heitjan. Weibull prediction of event times in clinical trials. Pharm Stat. 2008; 7:107-120.

Xiaoxi Zhang and Qi Long. Stochastic modeling and prediction for accrual in clinical trials. Stat in Med. 2010; 29:649-658.

Patrick Royston and Mahesh K. B. Parmar. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Stat in Med. 2002; 21:2175-2197.

`finalData` *Final enrollment and event data after achieving the target number of events*

Description

A data frame with 300 rows and 7 columns:

`trialsdt` The trial start date

`usubjid` The unique subject ID

`randdt` The randomization date

`treatment`	The treatment group number
`treatment_description`	Description of the treatment group
`time`	The day of event or censoring since randomization
`event`	The event indicator: 1 for event, 0 for non-event
`dropout`	The dropout indicator: 1 for dropout, 0 for non-dropout
`cutoffdt`	The cutoff date

For ongoing subjects, both `event` and `dropout` are equal to 0.

Usage

```
finalData
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 300 rows and 9 columns.

`fitDropout` *Fit time-to-dropout model*

Description

Fits a specified time-to-dropout model to the dropout data.

Usage

```
fitDropout(
  df,
  dropout_model = "exponential",
  piecewiseDropoutTime = 0,
  k_dropout = 0,
  scale_dropout = "hazard",
  m_dropout = 5,
  showplot = TRUE,
  by_treatment = FALSE,
  covariates = NULL,
  generate_plot = TRUE,
  interactive_plot = TRUE,
  nthreads = 0
)
```

Arguments

`df`	The subject-level dropout data, including `time` and `dropout`. The data should also include `treatment` coded as 1, 2, and so on, and `treatment_description` for fitting the dropout model by treatment.
`dropout_model`	The dropout model used to analyze the dropout data which can be set to one of the following options: "exponential", "Weibull", "log-logistic", "log-normal", "piecewise exponential", "model averaging", "spline", or "cox". The model averaging uses the `exp(-bic/2)` weighting and combines Weibull and log-normal models. The spline model of Royston and Parmar (2002) assumes that a transformation of the survival function is modeled as a natural cubic spline function of log time. By default, it is set to "exponential".
`piecewiseDropoutTime`	A vector that specifies the time intervals for the piecewise exponential dropout distribution. Must start with 0, e.g., c(0, 60) breaks the time axis into 2 event intervals: [0, 60) and [60, Inf). By default, it is set to 0.
`k_dropout`	The number of inner knots of the spline. The default `k_dropout=0` gives a Weibull, log-logistic or log-normal model, if `scale_dropout` is "hazard", "odds", or "normal", respectively. The knots are chosen as equally-spaced quantiles of the log uncensored survival times. The boundary knots are chosen as the minimum and maximum log uncensored survival times.
`scale_dropout`	The scale of the spline. The default is "hazard", in which case the log cumulative hazard is modeled as a spline function. If `scale = "odds"`, the log cumulative odds is modeled as a spline function. If `scale = "normal"`, `-qnorm(S(t))` is modeled as a spline function.
`m_dropout`	The number of dropout time intervals to extrapolate the hazard function beyond the last observed dropout time when `dropout_model = "cox"`.
`showplot`	A Boolean variable to control whether or not to show the fitted time-to-dropout survival curve. By default, it is set to `TRUE`.
`by_treatment`	A Boolean variable to control whether or not to fit the time-to-dropout data by treatment group. By default, it is set to `FALSE`.
`covariates`	The names of baseline covariates from the input data frame to include in the dropout model, e.g., c("age", "sex"). Factor variables need to be declared in the input data frame.
`generate_plot`	Whether to generate plots.
`interactive_plot`	Whether to produce interactive plots using plotly or static plots using ggplot2.
`nthreads`	Integer number of threads to use for ‘data.table’ (0 means the default data.table behavior).

Value

A list of results from the model fit including key information such as the dropout model, `model`, the estimated model parameters, `theta`, the covariance matrix, `vtheta`, as well as the Akaike Information Criterion, `aic`, and Bayesian Information Criterion, `bic`.

If the piecewise exponential model is used, the location of knots used in the model, `piecewiseDropoutTime`, will be included in the list of results.

If the model averaging option is chosen, the weight assigned to the Weibull component is indicated by the `w1` variable.

If the spline option is chosen, the `knots` and `scale` will be included in the list of results.

If the cox option is chosen, the list of results will include `model`, `theta`, `vtheta`, `aic`, `bic`, and `piecewiseDropoutTime`. Here

$$\theta = (\log(\lambda_1), \ldots, \log(\lambda_M), \beta^T)^T,$$

M denotes the number of distinct observed dropout times, $t_1 < \cdots < t_M$, λ_j denotes the estimated baseline hazard rate in the jth dropout time interval, $(t_{j-1}, t_j]$, and β represents the regression coefficients (log hazard ratios) from the Cox model. For a fair comparison, the estimation of baseline hazards is incorporated into the `aic` and `bic` values. In addition, piecewiseDropoutTime $=$ $(0, t_1, \ldots, t_M)$. To extend the survival curve beyond the last observed dropout time, a weighted average of the hazard rates from the final `m_dropout` dropout time intervals is used. The weights are proportional to the lengths of those intervals, i.e.,

$$\lambda_{M+1} = \sum_{j=M-m_{\text{dropout}}+1}^{M} w_j \lambda_j,$$

where $w_j = (t_j - t_{j-1})/(t_M - t_{M-m_{\text{dropout}}})$ for $j = M - m_{\text{dropout}} + 1, \ldots, M$.

When fitting the dropout model by treatment, the outcome is presented as a list of lists, where each list element corresponds to a specific treatment group.

The fitted time-to-dropout survival curve is also returned.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Patrick Royston and Mahesh K. B. Parmar. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Stat in Med. 2002; 21:2175-2197.

Examples

```
dropout_fit <- fitDropout(
  df = interimData2,
  dropout_model = "exponential",
  nthreads = 1)
```

`fitEnrollment` *Fit enrollment model*

Description

Fits a specified enrollment model to the enrollment data.

Usage

```
fitEnrollment(
  df,
  enroll_model = "b-spline",
  nknots = 0,
  accrualTime = 0,
  showplot = TRUE,
  generate_plot = TRUE,
  interactive_plot = TRUE,
  nthreads = 0
)
```

Arguments

`df`	The subject-level enrollment data, including `trialsdt`, `randdt` and `cutoffdt`.
`enroll_model`	The enrollment model which can be specified as "Poisson", "Time-decay", "B-spline", or "Piecewise Poisson". By default, it is set to "B-spline".
`nknots`	The number of inner knots for the B-spline enrollment model. By default, it is set to 0.
`accrualTime`	The accrual time intervals for the piecewise Poisson model. Must start with 0, e.g., c(0, 30) breaks the time axis into 2 accrual intervals: [0, 30) and [30, Inf). By default, it is set to 0.
`showplot`	A Boolean variable to control whether or not to show the fitted enrollment curve. By default, it is set to `TRUE`.
`generate_plot`	Whether to generate plots.
`interactive_plot`	Whether to produce interactive plots using plotly or static plots using ggplot2.
`nthreads`	Integer number of threads to use for 'data.table' (0 means the default data.table behavior).

Details

For the time-decay model, the mean function is

$$\mu(t) = (\mu/\delta)(t - (1/\delta)(1 - \exp(-\delta t)))$$

and the rate function is

$$\lambda(t) = (\mu/\delta)(1 - \exp(-\delta t)).$$

For the B-spline model, the daily enrollment rate is $\lambda(t) = \exp(B(t)'\theta)$, where $B(t)$ represents the B-spline basis functions.

Value

A list of results from the model fit including key information such as the enrollment model, `model`, the estimated model parameters, `theta`, the covariance matrix, `vtheta`, the Akaike Information Criterion, `aic`, and the Bayesian Information Criterion, `bic`, as well as the design matrix `x` for the B-spline enrollment model, and `accrualTime` for the piecewise Poisson enrollment model.

The fitted enrollment curve is also returned.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Xiaoxi Zhang and Qi Long. Stochastic modeling and prediction for accrual in clinical trials. Stat in Med. 2010; 29:649-658.

Examples

```
enroll_fit <- fitEnrollment(
  df = interimData1,
  enroll_model = "b-spline",
  nknots = 1,
  nthreads = 1)
```

`fitEvent` *Fit time-to-event model*

Description

Fits a specified time-to-event model to the event data.

Usage

```
fitEvent(
  df,
  event_model = "model averaging",
  piecewiseSurvivalTime = 0,
  k = 0,
  scale = "hazard",
  m = 5,
  showplot = TRUE,
  by_treatment = FALSE,
  covariates = NULL,
  generate_plot = TRUE,
  interactive_plot = TRUE,
  nthreads = 0
)
```

Arguments

`df`	The subject-level event data, including `time` and `event`. The data should also include `treatment` coded as 1, 2, and so on, and `treatment_description` for fitting the event model by treatment.
`event_model`	The event model used to analyze the event data which can be set to one of the following options: "exponential", "Weibull", "log-logistic", "log-normal", "piecewise exponential", "model averaging", "spline", or "cox". The model averaging uses the `exp(-bic/2)` weighting and combines Weibull and log-normal models. The spline model of Royston and Parmar (2002) assumes that a transformation of the survival function is modeled as a natural cubic spline function of log time. By default, it is set to "model averaging".
`piecewiseSurvivalTime`	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., c(0, 60) breaks the time axis into 2 event intervals: [0, 60) and [60, Inf). By default, it is set to 0.
`k`	The number of inner knots of the spline. The default `k=0` gives a Weibull, log-logistic or log-normal model, if `scale` is "hazard", "odds", or "normal", respectively. The knots are chosen as equally-spaced quantiles of the log uncensored survival times. The boundary knots are chosen as the minimum and maximum log uncensored survival times.
`scale`	The scale of the spline. The default is "hazard", in which case the log cumulative hazard is modeled as a spline function. If `scale = "odds"`, the log cumulative odds is modeled as a spline function. If `scale = "normal"`, `-qnorm(S(t))` is modeled as a spline function.
`m`	The number of event time intervals to extrapolate the hazard function beyond the last observed event time when `event_model = "cox"`.
`showplot`	A Boolean variable to control whether or not to show the fitted time-to-event survival curve. By default, it is set to `TRUE`.
`by_treatment`	A Boolean variable to control whether or not to fit the time-to-event data by treatment group. By default, it is set to `FALSE`.
`covariates`	The names of baseline covariates from the input data frame to include in the event model, e.g., c("age", "sex"). Factor variables need to be declared in the input data frame.
`generate_plot`	Whether to generate plots.
`interactive_plot`	Whether to produce interactive plots using plotly or static plots using ggplot2.
`nthreads`	Integer number of threads to use for ‘data.table’ (0 means the default data.table behavior).

Value

A list of results from the model fit including key information such as the event model, `model`, the estimated model parameters, `theta`, the covariance matrix, `vtheta`, as well as the Akaike Information Criterion, `aic`, and Bayesian Information Criterion, `bic`.

If the piecewise exponential model is used, the location of knots used in the model, `piecewiseSurvivalTime`, will be included in the list of results.

If the model averaging option is chosen, the weight assigned to the Weibull component is indicated by the `w1` variable.

If the spline option is chosen, the `knots` and `scale` will be included in the list of results.

If the cox option is chosen, the list of results will include `model`, `theta`, `vtheta`, `aic`, `bic`, and `piecewiseSurvivalTime`. Here

$$\theta = (\log(\lambda_1), \ldots, \log(\lambda_M), \beta^T)^T,$$

M denotes the number of distinct observed event times, $t_1 < \cdots < t_M$, λ_j denotes the estimated baseline hazard rate in the jth event time interval, $(t_{j-1}, t_j]$, and β represents the regression coefficients (log hazard ratios) from the Cox model. For a fair comparison, the estimation of baseline hazards is incorporated into the `aic` and `bic` values. In addition, $\text{piecewiseSurvivalTime} = (0, t_1, \ldots, t_M)$. To extend the survival curve beyond the last observed event time, a weighted average of the hazard rates from the final `m` event time intervals is used. The weights are proportional to the lengths of those intervals, i.e.,

$$\lambda_{M+1} = \sum_{j=M-m+1}^{M} w_j \lambda_j,$$

where $w_j = (t_j - t_{j-1})/(t_M - t_{M-m})$ for $j = M - m + 1, \ldots, M$.

When fitting the event model by treatment, the outcome is presented as a list of lists, where each list element corresponds to a specific treatment group.

The fitted time-to-event survival curve is also returned.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Patrick Royston and Mahesh K. B. Parmar. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Stat in Med. 2002; 21:2175-2197.

Examples

```
event_fit <- fitEvent(
  df = interimData2,
  event_model = "piecewise exponential",
  piecewiseSurvivalTime = c(0, 180),
  nthreads = 1)
```

`getPrediction` *Enrollment and event prediction*

Description

Performs enrollment and event prediction by utilizing observed data and specified enrollment and event models.

Usage

```
getPrediction(
  df = NULL,
  to_predict = "enrollment and event",
  target_n = NA,
  target_d = NA,
  enroll_model = "b-spline",
  nknots = 0,
  lags = 30,
  accrualTime = 0,
  enroll_prior = NULL,
  event_model = "model averaging",
  piecewiseSurvivalTime = 0,
  k = 0,
  scale = "hazard",
  m = 5,
  event_prior = NULL,
  dropout_model = "exponential",
  piecewiseDropoutTime = 0,
  k_dropout = 0,
  scale_dropout = "hazard",
  m_dropout = 5,
  dropout_prior = NULL,
  fixedFollowup = FALSE,
  followupTime = 365,
  pilevel = 0.9,
  nyears = 4,
  target_t = NA,
  nreps = 500,
  showEnrollment = TRUE,
  showEvent = TRUE,
  showDropout = FALSE,
  showOngoing = FALSE,
  showsummary = TRUE,
  showplot = TRUE,
  by_treatment = FALSE,
  ngroups = 1,
  alloc = NULL,
```

```
  treatment_label = NULL,
  covariates_event = NULL,
  event_prior_with_covariates = NULL,
  covariates_dropout = NULL,
  dropout_prior_with_covariates = NULL,
  fix_parameter = FALSE,
  generate_plot = TRUE,
  interactive_plot = TRUE,
  nthreads = 0
)
```

Arguments

`df`	The subject-level enrollment and event data, including `trialsdt`, `usubjid`, `randdt`, and `cutoffdt` for enrollment prediction, and, additionally, `time`, `event`, and `dropout` for event prediction. The data should also include `treatment` coded as 1, 2, and so on, and `treatment_description` for enrollment and event prediction by treatment. By default, it is set to `NULL` for enrollment and event prediction at the design stage.
`to_predict`	Specifies what to predict: "enrollment only", "event only", or "enrollment and event". By default, it is set to "enrollment and event".
`target_n`	The target number of subjects to enroll in the study.
`target_d`	The target number of events to reach in the study.
`enroll_model`	The enrollment model which can be specified as "Poisson", "Time-decay", "B-spline", or "Piecewise Poisson". By default, it is set to "B-spline".
`nknots`	The number of inner knots for the B-spline enrollment model. By default, it is set to 0.
`lags`	The day lags to compute the average enrollment rate to carry forward for the B-spline enrollment model. By default, it is set to 30.
`accrualTime`	The accrual time intervals for the piecewise Poisson model. Must start with 0, e.g., c(0, 30) breaks the time axis into 2 accrual intervals: [0, 30) and [30, Inf). By default, it is set to 0.
`enroll_prior`	The prior of enrollment model parameters.
`event_model`	The event model used to analyze the event data which can be set to one of the following options: "exponential", "Weibull", "log-logistic", "log-normal", "piecewise exponential", "model averaging", "spline", or "cox". The model averaging uses the `exp(-bic/2)` weighting and combines Weibull and log-normal models. By default, it is set to "model averaging".
`piecewiseSurvivalTime`	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., c(0, 60) breaks the time axis into 2 event intervals: [0, 60) and [60, Inf). By default, it is set to 0.
`k`	The number of inner knots of the spline event model of Royston and Parmar (2002). The default `k=0` gives a Weibull, log-logistic or log-normal model, if `scale` is "hazard", "odds", or "normal", respectively. The knots are chosen as equally-spaced quantiles of the log uncensored survival times. The boundary knots are chosen as the minimum and maximum log uncensored survival times.

`scale`	If "hazard", the log cumulative hazard is modeled as a spline function. If "odds", the log cumulative odds is modeled as a spline function. If "normal", -qnorm(S(t)) is modeled as a spline function.
`m`	The number of event time intervals to extrapolate the hazard function beyond the last observed event time.
`event_prior`	The prior of event model parameters.
`dropout_model`	The dropout model used to analyze the dropout data which can be set to one of the following options: "none", "exponential", "Weibull", "log-logistic", "log-normal", "piecewise exponential", "model averaging", "spline", or "cox". The model averaging uses the `exp(-bic/2)` weighting and combines Weibull and log-normal models. By default, it is set to "exponential".
`piecewiseDropoutTime`	A vector that specifies the time intervals for the piecewise exponential dropout distribution. Must start with 0, e.g., c(0, 60) breaks the time axis into 2 event intervals: [0, 60) and [60, Inf). By default, it is set to 0.
`k_dropout`	The number of inner knots of the spline dropout model of Royston and Parmar (2002). The default `k_dropout=0` gives a Weibull, log-logistic or log-normal model, if `scale_dropout` is "hazard", "odds", or "normal", respectively. The knots are chosen as equally-spaced quantiles of the log uncensored survival times. The boundary knots are chosen as the minimum and maximum log uncensored survival times.
`scale_dropout`	If "hazard", the log cumulative hazard for dropout is modeled as a spline function. If "odds", the log cumulative odds is modeled as a spline function. If "normal", -qnorm(S(t)) is modeled as a spline function.
`m_dropout`	The number of dropout time intervals to extrapolate the hazard function beyond the last observed dropout time.
`dropout_prior`	The prior of dropout model parameters.
`fixedFollowup`	A Boolean variable indicating whether a fixed follow-up design is used. By default, it is set to `FALSE` for a variable follow-up design.
`followupTime`	The follow-up time for a fixed follow-up design, in days. By default, it is set to 365.
`pilevel`	The prediction interval level. By default, it is set to 0.90.
`nyears`	The number of years after the data cut for prediction. By default, it is set to 4.
`target_t`	The target number of days after the data cutoff used to predict both the number of events and the probability of achieving the target event count.
`nreps`	The number of replications for simulation. By default, it is set to 500.
`showEnrollment`	A Boolean variable to control whether or not to show the number of enrolled subjects. By default, it is set to `TRUE`.
`showEvent`	A Boolean variable to control whether or not to show the number of events. By default, it is set to `TRUE`.
`showDropout`	A Boolean variable to control whether or not to show the number of dropouts. By default, it is set to `FALSE`.
`showOngoing`	A Boolean variable to control whether or not to show the number of ongoing subjects. By default, it is set to `FALSE`.

`showsummary`	A Boolean variable to control whether or not to show the prediction summary. By default, it is set to `TRUE`.
`showplot`	A Boolean variable to control whether or not to show the plots. By default, it is set to `TRUE`.
`by_treatment`	A Boolean variable to control whether or not to predict by treatment group. By default, it is set to `FALSE`.
`ngroups`	The number of treatment groups for enrollment prediction at the design stage. By default, it is set to 1. It is replaced with the actual number of treatment groups in the observed data if `df` is not `NULL`.
`alloc`	The treatment allocation in a randomization block. By default, it is set to `NULL`, which yields equal allocation among the treatment groups.
`treatment_label`	The treatment labels for treatments in a randomization block for design stage prediction. It is replaced with the treatment_description in the observed data if `df` is not `NULL`.
`covariates_event`	The names of baseline covariates from the input data frame to include in the event model, e.g., c("age", "sex"). Factor variables need to be declared in the input data frame.
`event_prior_with_covariates`	The prior of event model parameters in the presence of covariates.
`covariates_dropout`	The names of baseline covariates from the input data frame to include in the dropout model, e.g., c("age", "sex"). Factor variables need to be declared in the input data frame.
`dropout_prior_with_covariates`	The prior of dropout model parameters in the presence of covariates.
`fix_parameter`	Whether to fix parameters at the maximum likelihood estimates when generating new data for prediction. Defaults to `FALSE`, in which case, parameters will be drawn from their approximate posterior distribution.
`generate_plot`	Whether to generate plots.
`interactive_plot`	Whether to produce interactive plots using plotly or static plots using ggplot2.
`nthreads`	Integer number of threads to use for ‘data.table’ (0 means the default data.table behavior).

Details

For the time-decay model, the mean function is $\mu(t) = (\mu/\delta)(t - (1/\delta)(1 - \exp(-\delta t)))$ and the rate function is $\lambda(t) = (\mu/\delta)(1 - \exp(-\delta t))$. For the B-spline model, the daily enrollment rate is approximated as $\lambda(t) = \exp(B(t)'\theta)$, where `B(t)` represents the B-spline basis functions.

The `enroll_prior` variable should be a list that includes `model` to specify the enrollment model (poisson, time-decay, or piecewise poisson), `theta` and `vtheta` to indicate the parameter values and the covariance matrix. One can use a very small value of `vtheta` to fix the parameter values. For the piecewise Poisson enrollment model, the list should also include `accrualTime`. It should be noted that the B-spline model is not appropriate for use as prior.

For event prediction by treatment with prior information, the `event_prior` (`dropout_prior`) variable should be a list with one element per treatment. For each treatment, the element should include `model` to specify the event (dropout) model (exponential, weibull, log-logistic, log-normal, or piecewise exponential), and `theta` and `vtheta` to indicate the parameter values and the covariance matrix. For the piecewise exponential event (dropout) model, the list should also include `piecewiseSurvivalTime` (`piecewiseDropoutTime`) to indicate the location of knots. It should be noted that the model averaging, spline, and cox options are not appropriate for use as prior.

If the event prediction is not by treatment while the prior information is given by treatment, then each element of `event_prior` (`dropout_prior`) should also include `w` to specify the weight of the treatment in a randomization block. If the prediction is not by treatment and the prior is given for the overall study, then `event_prior` (`dropout_prior`) is a flat list with `model`, `theta`, and `vtheta`. For the piecewise exponential event (dropout) model, it should also include `piecewiseSurvivalTime` (`piecewiseDropoutTime`) to indicate the location of knots.

For analysis-stage enrollment and event prediction, the `enroll_prior`, `event_prior`, and `dropout_prior` are either set to `NULL` to use the observed data only, or specify the prior distribution of model parameters to be combined with observed data likelihood for enhanced modeling flexibility.

Value

A list containing model-fit objects and prediction objects.

The model-fit objects summarize either:

- the fitted models based on the observed data, or
- the posterior distribution of the model parameters when prior information is supplied.

The prediction objects may include:

- simulated enrollment data for future subjects, and
- simulated event data for both ongoing subjects and future subjects.

At the design stage, all predictions are based solely on prior information. In that case, the output includes `enroll_prior`, `event_prior`, and `dropout_prior`.

At the analysis stage, predictions are based on:

- the observed-data likelihood when no prior is provided, or
- the posterior distribution when prior information is provided.

When prior information is incorporated, the parameter vector `theta` in `enroll_post`, `event_post`, `event_post_with_covariates`, `dropout_post`, and `dropout_post_with_covariates` represents a weighted average of the prior mean and the maximum likelihood estimate. The corresponding variance-covariance matrix `vtheta` is the inverse of the total information matrix, where the total information is the sum of:

- the information from the prior distribution, and
- the information from the observed-data likelihood.

In addition to the model-fit objects, the output also includes the analysis stage at which prediction is performed, the prediction target, and the enrollment and event prediction results when applicable.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Event prediction after enrollment completion
set.seed(3000)

pred <- getPrediction(
  df = interimData2,
  to_predict = "event only",
  target_d = 200,
  event_model = "weibull",
  dropout_model = "exponential",
  pilevel = 0.90,
  nreps = 100,
  nthreads = 1)
```

`interimData1` *Interim enrollment and event data before enrollment completion*

Description

A data frame with 225 rows and 9 columns:

`trialsdt` The trial start date

`usubjid` The unique subject ID

`randdt` The randomization date

`treatment` The treatment group number

`treatment_description` Description of the treatment group

`time` The day of event or censoring since randomization

`event` The event indicator: 1 for event, 0 for non-event

`dropout` The dropout indicator: 1 for dropout, 0 for non-dropout

`cutoffdt` The cutoff date

For ongoing subjects, both `event` and `dropout` are equal to 0.

Usage

```
interimData1
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 224 rows and 9 columns.

`interimData2` *Interim enrollment and event data after enrollment completion*

Description

A data frame with 300 rows and 7 columns:

`trialsdt`	The trial start date
`usubjid`	The unique subject ID
`randdt`	The randomization date
`treatment`	The treatment group number
`treatment_description`	Description of the treatment group
`time`	The day of event or censoring since randomization
`event`	The event indicator: 1 for event, 0 for non-event
`dropout`	The dropout indicator: 1 for dropout, 0 for non-dropout
`cutoffdt`	The cutoff date

For ongoing subjects, both `event` and `dropout` are equal to 0.

Usage

```
interimData2
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 300 rows and 9 columns.

`predictEnrollment` *Predict enrollment*

Description

Utilizes a pre-fitted enrollment model to generate enrollment times for new subjects and provide a prediction interval for the expected time to reach the enrollment target.

Usage

```
predictEnrollment(
  df = NULL,
  target_n = NA,
  enroll_fit = NULL,
  lags = 30,
  pilevel = 0.9,
  nyears = 4,
  nreps = 500,
  showsummary = TRUE,
  showplot = TRUE,
  by_treatment = FALSE,
  ngroups = 1,
  alloc = NULL,
  treatment_label = NULL,
  fix_parameter = FALSE,
  generate_plot = TRUE,
  interactive_plot = TRUE,
  nthreads = 0
)
```

Arguments

`df`	The subject-level enrollment data, including `trialsdt`, `randdt` and `cutoffdt`. The data should also include `treatment` coded as 1, 2, and so on, and `treatment_description` for prediction by treatment group. By default, it is set to `NULL` for enrollment prediction at the design stage.
`target_n`	The target number of subjects to enroll in the study.
`enroll_fit`	The pre-fitted enrollment model used to generate predictions.
`lags`	The day lags to compute the average enrollment rate to carry forward for the B-spline enrollment model. By default, it is set to 30.
`pilevel`	The prediction interval level. By default, it is set to 0.90.
`nyears`	The number of years after the data cut for prediction. By default, it is set to 4.
`nreps`	The number of replications for simulation. By default, it is set to 500.
`showsummary`	A Boolean variable to control whether or not to show the prediction summary. By default, it is set to `TRUE`.
`showplot`	A Boolean variable to control whether or not to show the prediction plot. By default, it is set to `TRUE`.
`by_treatment`	A Boolean variable to control whether or not to predict enrollment by treatment group. By default, it is set to `FALSE`.
`ngroups`	The number of treatment groups for enrollment prediction at the design stage. By default, it is set to 1. It is replaced with the actual number of treatment groups in the observed data if `df` is not `NULL`.
`alloc`	The treatment allocation in a randomization block. By default, it is set to `NULL`, which yields equal allocation among the treatment groups.

`treatment_label`	The treatment labels for treatments in a randomization block for design stage prediction. It is replaced with the treatment_description in the observed data if `df` is not `NULL`.
`fix_parameter`	Whether to fix parameters at the maximum likelihood estimates when generating new data for prediction. Defaults to FALSE, in which case, parameters will be drawn from their approximate posterior distributions.
`generate_plot`	Whether to generate plots.
`interactive_plot`	Whether to produce interactive plots using plotly or static plots using ggplot2.
`nthreads`	Integer number of threads to use for 'data.table' (0 means the default data.table behavior).

Details

The `enroll_fit` variable can be used for enrollment prediction at the design stage. A piecewise Poisson model can be parameterized through the time intervals, `accrualTime`, which is treated as fixed, and the enrollment rates in the intervals, `accrualIntensity`, the log of which is used as the model parameter. For the homogeneous Poisson, time-decay, and piecewise Poisson models, `enroll_fit` is used to specify the prior distribution of model parameters, with a very small variance being used to fix the parameter values. It should be noted that the B-spline model is not appropriate for use during the design stage.

During the enrollment stage, `enroll_fit` is the enrollment model fit based on the observed data. The fitted enrollment model is used to generate enrollment times for new subjects.

Value

A list of prediction results, which includes important information such as the median, lower and upper percentiles for the estimated time to reach the target number of subjects, as well as simulated enrollment data for new subjects. The data for the prediction plot is also included within the list.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Xiaoxi Zhang and Qi Long. Stochastic modeling and prediction for accrual in clinical trials. Stat in Med. 2010; 29:649-658.

Examples

```
# Enrollment prediction at the design stage
set.seed(1000)

enroll_pred <- predictEnrollment(
  target_n = 300,
  enroll_fit = list(
    model = "piecewise poisson",
```

```
    theta = log(26/9*seq(1, 9)/30.4375),
    vtheta = diag(9)*1e-8,
    accrualTime = seq(0, 8)*30.4375),
  pilevel = 0.90,
  nreps = 100,
  nthreads = 1)
```

---

predictEvent *Predict event*

Description

Utilizes pre-fitted time-to-event and time-to-dropout models to generate event and dropout times for ongoing subjects and new subjects. It also provides a prediction interval for the expected time to reach the target number of events.

Usage

```
predictEvent(
  df = NULL,
  target_d = NA,
  newSubjects = NULL,
  event_fit = NULL,
  m = 5,
  dropout_fit = NULL,
  m_dropout = 5,
  fixedFollowup = FALSE,
  followupTime = 365,
  pilevel = 0.9,
  nyears = 4,
  target_t = NA,
  nreps = 500,
  showEnrollment = TRUE,
  showEvent = TRUE,
  showDropout = FALSE,
  showOngoing = FALSE,
  showsummary = TRUE,
  showplot = TRUE,
  by_treatment = FALSE,
  covariates_event = NULL,
  event_fit_with_covariates = NULL,
  covariates_dropout = NULL,
  dropout_fit_with_covariates = NULL,
  fix_parameter = FALSE,
  generate_plot = TRUE,
  interactive_plot = TRUE,
  nthreads = 0
)
```

Arguments

`df`	The subject-level enrollment and event data, including `trialsdt`, `usubjid`, `randdt`, `cutoffdt`, `time`, `event`, and `dropout`. The data should also include `treatment` coded as 1, 2, and so on, and `treatment_description` for by-treatment prediction. By default, it is set to `NULL` for event prediction at the design stage.
`target_d`	The target number of events to reach in the study.
`newSubjects`	The enrollment data for new subjects including `draw` and `arrivalTime`. The data should also include `treatment` for prediction by treatment. By default, it is set to `NULL`, indicating the completion of subject enrollment.
`event_fit`	The pre-fitted event model used to generate predictions.
`m`	The number of event time intervals to extrapolate the hazard function beyond the last observed event time.
`dropout_fit`	The pre-fitted dropout model used to generate predictions. By default, it is set to `NULL`, indicating no dropout.
`m_dropout`	The number of dropout time intervals to extrapolate the hazard function beyond the last observed dropout time.
`fixedFollowup`	A Boolean variable indicating whether a fixed follow-up design is used. By default, it is set to `FALSE` for a variable follow-up design.
`followupTime`	The follow-up time for a fixed follow-up design, in days. By default, it is set to 365.
`pilevel`	The prediction interval level. By default, it is set to 0.90.
`nyears`	The number of years after the data cut for prediction. By default, it is set to 4.
`target_t`	The target number of days after the data cutoff used to predict both the number of events and the probability of achieving the target event count.
`nreps`	The number of replications for simulation. By default, it is set to 500. If `newSubjects` is not `NULL`, the number of draws in `newSubjects` should be `nreps`.
`showEnrollment`	A Boolean variable to control whether or not to show the number of enrolled subjects. By default, it is set to `TRUE`.
`showEvent`	A Boolean variable to control whether or not to show the number of events. By default, it is set to `TRUE`.
`showDropout`	A Boolean variable to control whether or not to show the number of dropouts. By default, it is set to `FALSE`.
`showOngoing`	A Boolean variable to control whether or not to show the number of ongoing subjects. By default, it is set to `FALSE`.
`showsummary`	A Boolean variable to control whether or not to show the prediction summary. By default, it is set to `TRUE`.
`showplot`	A Boolean variable to control whether or not to show the prediction plot. By default, it is set to `TRUE`.
`by_treatment`	A Boolean variable to control whether or not to predict event by treatment group. By default, it is set to `FALSE`.

`covariates_event`	The names of baseline covariates from the input data frame to include in the event model, e.g., c("age", "sex"). Factor variables need to be declared in the input data frame.
`event_fit_with_covariates`	The pre-fitted event model with covariates used to generate event predictions for ongoing subjects.
`covariates_dropout`	The names of baseline covariates from the input data frame to include in the dropout model, e.g., c("age", "sex"). Factor variables need to be declared in the input data frame.
`dropout_fit_with_covariates`	The pre-fitted dropout model with covariates used to generate dropout predictions for ongoing subjects.
`fix_parameter`	Whether to fix parameters at the maximum likelihood estimates when generating new data for prediction. Defaults to FALSE, in which case, parameters will be drawn from their approximate posterior distribution.
`generate_plot`	Whether to generate plots.
`interactive_plot`	Whether to produce interactive plots using plotly or static plots using ggplot2.
`nthreads`	Integer number of threads to use for ‘data.table’ (0 means the default data.table behavior).

Details

To ensure successful event prediction at the design stage, it is important to provide the `newSubjects` data set.

To specify the event (dropout) model used during the design-stage event prediction, the `event_fit` (`dropout_fit`) should be a list with one element per treatment. For each treatment, the element should include `model` to specify the event model (exponential, weibull, log-logistic, log-normal, or piecewise exponential), and `theta` and `vtheta` to indicate the parameter values and the covariance matrix. For the piecewise exponential event (dropout) model, the list should also include `piecewiseSurvivalTime` (`piecewiseDropoutTime`) to indicate the location of knots. It should be noted that the model averaging and spline options are not appropriate for use during the design stage.

Following the commencement of the trial, we obtain the event model fit and the dropout model fit based on the observed data, denoted as `event_fit` and `dropout_fit`, respectively. These fitted models are subsequently utilized to generate event and dropout times for both ongoing and new subjects in the trial.

Value

A list of prediction results which includes important information such as the median, lower and upper percentiles for the estimated day and date to reach the target number of events, as well as simulated event data for both ongoing and new subjects. The data for the prediction plot is also included within this list. If target_t is specified, it additionally provides the median, lower, and upper percentiles of the event count at target_t, as well as the predictive probability of achieving the target number of events by target_t.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Emilia Bagiella and Daniel F. Heitjan. Predicting analysis times in randomized clinical trials. Stat in Med. 2001; 20:2055-2063.

Gui-shuang Ying and Daniel F. Heitjan. Weibull prediction of event times in clinical trials. Pharm Stat. 2008; 7:107-120.

Examples

```
# Event prediction after enrollment completion
set.seed(2000)

event_fits <- fitEvent(
  df = interimData2,
  event_model = "piecewise exponential",
  piecewiseSurvivalTime = c(0, 140, 352),
  nthreads = 1)

dropout_fits <- fitDropout(
  df = interimData2,
  dropout_model = "exponential",
  nthreads = 1)

event_pred <- predictEvent(
  df = interimData2,
  target_d = 200,
  event_fit = event_fits$fit,
  dropout_fit = dropout_fits$fit,
  pilevel = 0.90,
  nreps = 100,
  nthreads = 1)
```

`pwexpreg` *Piecewise exponential regression*

Description

Obtains the maximum likelihood estimates for piecewise exponential regression.

Usage

```
pwexpreg(time, event, J, tcut, q = 0, x = 1)
```

Arguments

`time`	The survival time.
`event`	The event indicator.
`J`	The number of time intervals.
`tcut`	A vector that specifies the endpoints of time intervals for the baseline piecewise exponential survival distribution. Must start with 0, e.g., c(0, 60) breaks the time axis into 2 event intervals: [0, 60) and [60, Inf). By default, it is set to 0.
`q`	The number of columns of the covariates matrix (exluding the intercept).
`x`	The covariates matrix (including the intercept).

Value

The maximum likelihood estimates and the associated covariance matrix, AIC and BIC.

`runShinyApp_eventPred` *Run Shiny app*

Description

Runs the event prediction Shiny app.

Usage

```
runShinyApp_eventPred()
```

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

`summarizeObserved` *Summarize observed data*

Description

Provides an overview of the observed data, including the trial start date, data cutoff date, enrollment duration, number of subjects enrolled, number of events and dropouts, number of subjects at risk, cumulative enrollment and event data, daily enrollment rates, and Kaplan-Meier plots for time to event and time to dropout.

Usage

```
summarizeObserved(
  df,
  to_predict = "event only",
  showplot = TRUE,
  by_treatment = FALSE,
  generate_plot = TRUE,
  interactive_plot = TRUE,
  nthreads = 0
)
```

Arguments

`df`	The subject-level data, including `trialsdt`, `usubjid`, `randdt`, and `cutoffdt` for enrollment prediction, as well as `time`, `event` and `dropout` for event prediction, and `treatment` coded as 1, 2, and so on, and `treatment_description` for prediction by treatment group.
`to_predict`	Specifies what to predict: "enrollment only", "event only", or "enrollment and event". By default, it is set to "event only".
`showplot`	A Boolean variable to control whether or not to show the observed data plots. By default, it is set to `TRUE`.
`by_treatment`	A Boolean variable to control whether or not to summarize observed data by treatment group. By default, it is set to `FALSE`.
`generate_plot`	Whether to generate plots.
`interactive_plot`	Whether to produce interactive plots using plotly or static plots using ggplot2.
`nthreads`	Integer number of threads to use for ‘data.table’ (0 means the default data.table behavior).

Value

A list that includes a range of summary statistics, data sets, and plots depending on the value of `to_predict`.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
observed1 <- summarizeObserved(
  df = interimData1,
  to_predict = "enrollment and event",
  nthreads = 1)

observed2 <- summarizeObserved(
  df = interimData2,
  to_predict = "event only",
```

```
    nthreads = 1)
```

Index