Package: DTEAssurance (via r-universe)

June 8, 2026

Type Package

Title Assurance Methods for Clinical Trials with a Delayed Treatment Effect

Version 1.1.0

Description Provides functions for planning clinical trials subject to a delayed treatment effect using assurance-based methods. Includes two 'shiny' applications for interactive exploration, simulation, and visualisation of trial designs and outcomes. The methodology is described in: Salsbury JA, Oakley JE, Julious SA, Hampson LV (2024) ``Assurance methods for designing a clinical trial with a delayed treatment effect'' <doi:10.1002/sim.10136>, Salsbury JA, Oakley JE, Julious SA, Hampson LV (2024) ``Adaptive clinical trial design with delayed treatment effects using elicited prior distributions'' <doi:10.48550/arXiv.2509.07602>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports SHELF, stats, survival, nleqslv, nph, nphRCT, dplyr, rpact, magrittr, rlang, future.apply

Suggests testthat (>= 3.0.0), rjags, shiny, shinyAce, shinyBS, shinyjs, rhandsontable, plotly

Config/testthat/edition 3

Depends R (>= 4.0)

Date 2025-12-02

URL `https://jamesalsbury.github.io/DTEAssurance/`

Config/pak/sysreqs
cmake make libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libssl-dev zlib1g-dev

Repository https://jamesalsbury.r-universe.dev

Date/Publication 2026-05-09 11:30:02 UTC

RemoteUrl https://github.com/jamesalsbury/dteassurance

RemoteRef HEAD

RemoteSha 8936ad766168fe14241533909a04d5b8672e4216

Contents

`add_recruitment_time` *Add recruitment time to a survival dataset*

Description

Simulates recruitment timing for each patient in a survival dataset using either a power model or a piecewise constant (PWC) model. The function appends recruitment times and pseudo survival times (time from recruitment to event or censoring).

Usage

```
add_recruitment_time(
  data,
  rec_method,
  rec_period = NULL,
  rec_power = NULL,
  rec_rate = NULL,
  rec_duration = NULL
)
```

Arguments

`data`	A dataframe containing survival data with columns: `time`, `status`, and `group`
`rec_method`	Recruitment method: `"power"` for power model or `"PWC"` for piecewise constant model
`rec_period`	Period length for the power model
`rec_power`	Power parameter for the power model
`rec_rate`	Comma-separated string of recruitment rates for the PWC model
`rec_duration`	Comma-separated string of durations corresponding to each rate in the PWC model

Value

A dataframe with two additional columns:

rec_time Simulated recruitment time for each patient

pseudo_time Time from recruitment to event or censoring

Class: `data.frame`

Examples

```
set.seed(123)
df <- data.frame(
  time = rexp(20, rate = 0.1),
  status = rbinom(20, 1, 0.8),
  group = rep(c("Control", "Treatment"), each = 10)
)
recruited <- add_recruitment_time(df, rec_method = "power", rec_period = 12, rec_power = 1)
head(recruited)
```

`assurance_adaptive_shiny_app`

Launch the 'shiny' adaptive assurance app

Description

Launches a 'shiny' application to simulate group sequential trials with delayed treatment effects (DTE) using elicited prior distributions. The app allows interactive exploration of trial designs and assurance calculations.

Usage

```
assurance_adaptive_shiny_app()
```

Value

No return value, called for side effects (invisibly returns NULL). The function launches an interactive 'shiny' application.

Examples

```
# Launch the interactive Shiny app
assurance_adaptive_shiny_app()
```

`assurance_shiny_app` *Launch the 'shiny' Assurance app*

Description

Launches a 'shiny' application to calculate assurance for clinical trials where delayed treatment effects (DTE) may be present. The app allows elicitation of prior distributions and calculates assurance metrics.

Usage

```
assurance_shiny_app()
```

Value

No return value, called for side effects (invisibly returns NULL). The function launches an interactive 'shiny' application.

Examples

```
# Launch the interactive Shiny app
assurance_shiny_app()
```

`BPP_func` *Calculate Bayesian Predictive Probability given interim data and posterior samples*

Description

Calculate Bayesian Predictive Probability given interim data and posterior samples

Usage

```
BPP_func(
  data,
  posterior_df,
  control_distribution = "Exponential",
  n_c_planned,
  n_t_planned,
  rec_time_planned,
  df_cens_time,
  censoring_model,
  analysis_model,
  n_sims = 500
)
```

Arguments

`data` A data frame containing interim survival data, censored at `df_cens_time`, with columns:

- `time` Final observed/event time at the interim (on the analysis time scale).
- `group` Treatment group indicator (e.g. "Control", "Treatment").
- `rec_time` Recruitment (calendar) time.
- `pseudo_time` `time` + `rec_time` (calendar time at event/censoring).
- `status` Event indicator at the interim (1 = event, 0 = censored).
- `survival_time` Observed follow-up time from randomisation to event/censoring at the interim.

`posterior_df` A data frame of posterior samples with columns: `lambda_c`, `delay_time` and `HR`, corresponding to the control hazard, the delay (changepoint) time and the post-delay hazard ratio, respectively.

`control_distribution` Distributional form assumed for the control arm: either `"Exponential"` (default) or `"Weibull"`.

`n_c_planned` Planned maximum number of patients in the control group.

`n_t_planned` Planned maximum number of patients in the treatment group.

`rec_time_planned` Planned maximum recruitment calendar time for the full trial.

`df_cens_time` Calendar time at which `df` has been censored (interim analysis time).

`censoring_model` A named list specifying the censoring mechanism for the future data:

- `method`: one of `"Time"`, `"Events"`, or `"IF"`.
- `time`, `events`, `IF`: parameters for the corresponding method.

`analysis_model` A named list specifying the final analysis and decision rule:

- `method`: e.g. `"LRT"`, `"WLRT"`, or `"MW"`.
- `alpha`: one-sided type I error level.
- `alternative_hypothesis`: direction of the alternative (e.g. `"one.sided"`).

- `rho`, `gamma`, `t_star`, `s_star`: additional parameters for WLRT or MW (if applicable).

`n_sims` Number of predictive simulations to run (default is 1000).

Value

A single numeric value giving the Bayesian predictive probability of success at the final analysis under the specified design, censoring model and analysis model.

Examples

```
set.seed(123)
n <- 30
cens_time <- 15

time <- runif(n, 0, 12)
rec_time <- runif(n, 0, 12)


df <- data.frame(
  time = time,
  group = c(rep("Control", n/2), rep("Treatment", n/2)),
  rec_time = rec_time
)


df$pseudo_time <- df$time + df$rec_time
df$status <- df$pseudo_time < cens_time
df$survival_time <- ifelse(df$status == TRUE, df$time, cens_time - df$rec_time)


posterior_df <- data.frame(HR = rnorm(20, mean = 0.75, sd = 0.05),
                           delay_time = rep(0, 20),
                           lambda_c = rnorm(20, log(2)/9, sd = 0.01))


censoring_model = list(method = "Time", time = 25)
analysis_model = list(method = "LRT",
                      alpha = 0.025,
                      alternative_hypothesis = "one.sided")

BPP_outcome <- BPP_func(df,
           posterior_df,
           control_distribution = "Exponential",
           n_c_planned = n/2,
           n_t_planned = n/2,
           rec_time_planned = 12, df_cens_time = 15,
           censoring_model = censoring_model,
           analysis_model = analysis_model,
           n_sims = 10)
```

calc_dte_assurance — *Calculate Assurance for a Trial with a Delayed Treatment Effect*

Description

Simulates operating characteristics for a clinical trial under prior uncertainty about a delayed treatment effect. The function integrates beliefs about control survival, treatment delay, post-delay hazard ratio, recruitment, censoring, and analysis method to estimate assurance and other trial metrics.

Usage

```
calc_dte_assurance(
  n_c,
  n_t,
  control_model,
  effect_model,
  censoring_model,
  recruitment_model,
  analysis_model,
  n_sims = 1000
)
```

Arguments

`n_c` Vector of control group sample sizes

`n_t` Vector of treatment group sample sizes

`control_model` A named list specifying the control arm survival distribution:

- `dist`: Distribution type ("Exponential" or "Weibull")
- `parameter_mode`: Either "Fixed" or "Distribution"
- `fixed_type`: If "Fixed", specify as "Parameters" or "Landmark"
- `lambda`, `gamma`: Scale and shape parameters
- `t1`, `t2`: Landmark times
- `surv_t1`, `surv_t2`: Survival probabilities at landmarks
- `t1_Beta_a`, `t1_Beta_b`, `diff_Beta_a`, `diff_Beta_b`: Beta prior parameters

`effect_model` A named list specifying beliefs about the treatment effect:

- `delay_SHELF`, `HR_SHELF`: SHELF objects encoding beliefs
- `delay_dist`, `HR_dist`: Distribution types ("hist" by default)
- `P_S`: Probability that survival curves separate
- `P_DTE`: Probability of delayed separation, conditional on separation

`censoring_model`

A named list specifying the censoring mechanism:

- `method`: "Time", "Events", or "IF"

- `time`, `events`, `IF`: Parameters for each method

`recruitment_model`
A named list specifying the recruitment process:

- `method`: "power" or "PWC"
- `period`, `power`: Parameters for power model
- `rate`, `duration`: Comma-separated strings for PWC model

`analysis_model` A named list specifying the statistical test and decision rule:

- `method`: "LRT", "WLRT", or "MW"
- `alpha`, `alternative_hypothesis`: Type I error and hypothesis direction
- `rho`, `gamma`, `t_star`, `s_star`: Parameters for WLRT or MW
- `success_threshold_HR`: Optional threshold for declaring success

`n_sims` Number of simulations to run (default = 1000)

Value

A named list containing:

assurance Estimated assurance (probability of success under prior uncertainty)

CI 95% confidence interval for assurance

duration Mean trial duration across simulations

sample_size Mean sample size across simulations

diagnostics Additional diagnostics if `success_threshold_HR` is specified

Class: `list`

Examples

```
set.seed(123)
control_model <- list(dist = "Exponential", parameter_mode = "Fixed",
fixed_type = "Parameters", lambda = 0.1)
effect_model <- list(delay_SHELF = SHELF::fitdist(c(3, 4, 5),
probs = c(0.25, 0.5, 0.75), lower = 0, upper = 10),
delay_dist = "gamma",
HR_SHELF = SHELF::fitdist(c(0.55, 0.6, 0.7), probs = c(0.25, 0.5, 0.75), lower = 0, upper = 1.5),
HR_dist = "gamma",
P_S = 1, P_DTE = 0)
censoring_model <- list(method = "Time", time = 12)
recruitment_model <- list(method = "power", period = 12, power = 1)
analysis_model <- list(method = "LRT", alpha = 0.025, alternative_hypothesis = "two.sided")
result <- calc_dte_assurance(n_c = 300, n_t = 300,
                                     control_model = control_model,
                                     effect_model = effect_model,
                                     censoring_model = censoring_model,
                                     recruitment_model = recruitment_model,
                                     analysis_model = analysis_model,
                                     n_sims = 10)
str(result)
```

`calc_dte_assurance_adaptive`
Calculates operating characteristics for a Group Sequential Trial with a Delayed Treatment Effect

---

Description

Simulates assurance and operating characteristics for a group sequential trial under prior uncertainty about a delayed treatment effect. The function integrates beliefs about control survival, treatment delay, post-delay hazard ratio, recruitment, and group sequential design (GSD) parameters.

Usage

```
calc_dte_assurance_adaptive(
  n_c,
  n_t,
  control_model,
  effect_model,
  recruitment_model,
  GSD_model,
  analysis_model = NULL,
  n_sims = 1000
)
```

Arguments

`n_c` Control group sample size

`n_t` Treatment group sample size

`control_model` A named list specifying the control arm survival distribution:

- `dist`: Distribution type ("Exponential" or "Weibull")
- `parameter_mode`: Either "Fixed" or "Distribution"
- `fixed_type`: If "Fixed", specify as "Parameters" or "Landmark"
- `lambda`, `gamma`: Scale and shape parameters
- `t1`, `t2`: Landmark times
- `surv_t1`, `surv_t2`: Survival probabilities at landmarks
- `t1_Beta_a`, `t1_Beta_b`, `diff_Beta_a`, `diff_Beta_b`: Beta prior parameters

`effect_model` A named list specifying beliefs about the treatment effect:

- `delay_SHELF`, `HR_SHELF`: SHELF objects encoding beliefs
- `delay_dist`, `HR_dist`: Distribution types ("hist" by default)
- `P_S`: Probability that survival curves separate
- `P_DTE`: Probability of delayed separation, conditional on separation

`recruitment_model`
A named list specifying the recruitment process:

- method: "power" or "PWC"
- period, power: Parameters for power model
- rate, duration: Comma-separated strings for PWC model

GSD_model A named list specifying the group sequential design:
- events: Total number of events
- alpha_spending: Cumulative alpha spending vector
- alpha_IF: Information Fraction at which we look for efficacy
- futility_type: beta (for beta-spending), BPP (for Bayesian Predictive Probability) or none
- futility_IF: Information Fraction at which we look for futility
- beta_spending: Cumulative beta spending vector
- BPP_threshold: BPP value at which we will stop for futility

analysis_model A named list specifying the final analysis and decision rule:
- method: e.g. "LRT", "WLRT", or "MW".
- alpha: one-sided type I error level.
- alternative_hypothesis: direction of the alternative (e.g. "one.sided").
- rho, gamma, t_star, s_star: additional parameters for WLRT or MW (if applicable).

n_sims Number of simulations to run (default = 1000)

Value

A data frame with one row per simulated trial and the following columns:

Trial Simulation index

IF Information fraction label used at the decision point

Decision Interim decision outcome (e.g., "Continue", "Stop for efficacy", "Stop for futility")

StopTime Time at which the trial stopped or completed

SampleSize Total sample size at the time of decision

Final_Decision Final classification of trial success based on the test statistic and threshold

Class: data.frame

Examples

```
set.seed(123)
control_model <- list(dist = "Exponential", parameter_mode = "Fixed",
fixed_type = "Parameters", lambda = 0.1)
effect_model <- list(P_S = 1, P_DTE = 0,
HR_SHELF = SHELF::fitdist(c(0.6, 0.65, 0.7), probs = c(0.25, 0.5, 0.75), lower = 0, upper = 2),
HR_dist = "gamma",
delay_SHELF = SHELF::fitdist(c(3, 4, 5), probs = c(0.25, 0.5, 0.75), lower = 0, upper = 10),
delay_dist = "gamma"
)
recruitment_model <- list(method = "power", period = 12, power = 1)
GSD_model <- list(events = 300, alpha_spending = c(0.0125, 0.025),
```

```
                    alpha_IF = c(0.75, 1), futility_type = "none")
result <- calc_dte_assurance_adaptive(n_c = 300, n_t = 300,
                           control_model = control_model,
                           effect_model = effect_model,
                           recruitment_model = recruitment_model,
                           GSD_model = GSD_model,
                           n_sims = 10)
str(result)
```

---

calibrate_BPP_threshold *Function to calculate the 'optimal' BPP threshold value*

Description

Function to calculate the 'optimal' BPP threshold value

Usage

```
calibrate_BPP_threshold(
  n_c,
  n_t,
  control_model,
  effect_model,
  recruitment_model,
  IA_model,
  analysis_model,
  data_generating_model,
  n_df_sims = 100,
  update_priors_sims = 1000,
  PP_sims = 2000
)
```

Arguments

`n_c`	Number of control patients
`n_t`	Number of treatment patients
`control_model`	A named list specifying the control arm survival distribution:

- `dist`: Distribution type ("Exponential" or "Weibull")
- `parameter_mode`: Either "Fixed" or "Distribution"
- `fixed_type`: If "Fixed", specify as "Parameters" or "Landmark"
- `lambda`, `gamma`: Scale and shape parameters
- `t1`, `t2`: Landmark times
- `surv_t1`, `surv_t2`: Survival probabilities at landmarks

- t1_Beta_a, t1_Beta_b, diff_Beta_a, diff_Beta_b: Beta prior parameters

effect_model
: A named list specifying beliefs about the treatment effect:
- delay_SHELF, HR_SHELF: SHELF objects encoding beliefs
- delay_dist, HR_dist: Distribution types ("hist" by default)
- P_S: Probability that survival curves separate
- P_DTE: Probability of delayed separation, conditional on separation

recruitment_model
: A named list specifying the recruitment process:
- method: "power" or "PWC"
- period, power: Parameters for power model
- rate, duration: Comma-separated strings for PWC model

IA_model
: A named list specifying the censoring mechanism for the future data:
- events: Number of events which is 100% information fraction
- IF: The information fraction at which to censor and calculate BPP

analysis_model
: A named list specifying the final analysis and decision rule:
- method: e.g. "LRT", "WLRT", or "MW".
- alpha: one-sided type I error level.
- alternative_hypothesis: direction of the alternative (e.g. "one.sided").
- rho, gamma, t_star, s_star: additional parameters for WLRT or MW (if applicable).

data_generating_model
: A named list specifying the parameters for the data-generating mechanism
- lambda_c: hazard rate for the control group
- delay_time: time at which the treatment starts to take effect
- post_delay_HR: hazard ratio, after delay_time

n_df_sims
: Number of data sets to simulate (default is 100).

update_priors_sims
: Number of samples to generate from the posterior (default is 1000)

PP_sims
: Number of simulations used to calculate the predictive probability (default is 2000)

Value

A vector of length n_sims corresponding to the value of BPP for each simulated trial

Examples

```
set.seed(123)
control_model = list(dist = "Exponential",
                     parameter_mode = "Distribution",
                     t1 = 12,
                     t1_Beta_a = 20,
                     t1_Beta_b = 32)
```

```
effect_model = list(delay_SHELF = SHELF::fitdist(c(5.5, 6, 6.5),
                    probs = c(0.25, 0.5, 0.75), lower = 0, upper = 12),
                    delay_dist = "gamma",
                    HR_SHELF = SHELF::fitdist(c(0.5, 0.6, 0.7),
                    probs = c(0.25, 0.5, 0.75), lower = 0, upper = 1),
                    HR_dist = "gamma",
                    P_S = 1,
                    P_DTE = 0)

recruitment_model <- list(method = "power", period = 12, power = 1)

IA_model = list(events = 20, IF = 0.5)

analysis_model = list(method = "LRT",
                      alpha = 0.025,
                      alternative_hypothesis = "one.sided")


data_generating_model = list(lambda_c = log(2)/12,
                             gamma_c = NULL,
                             delay_time = 3,
                             post_delay_HR = 0.75)


threshold <- calibrate_BPP_threshold(n_c = 15, n_t = 15,
                     control_model = control_model,
                     effect_model = effect_model,
                     recruitment_model = recruitment_model,
                     IA_model = IA_model,
                     analysis_model = analysis_model,
                     data_generating_model = data_generating_model,
                     n_df_sims = 2)
```

`calibrate_BPP_timing` *Function to calculate the 'optimal' information fraction to calculate BPP*

Description

Function to calculate the 'optimal' information fraction to calculate BPP

Usage

```
calibrate_BPP_timing(
  n_c,
  n_t,
  control_model,
  effect_model,
```

```
  recruitment_model,
  IA_model,
  analysis_model,
  n_sims = 50
)
```

Arguments

n_c Number of control patients

n_t Number of treatment patients

control_model A named list specifying the control arm survival distribution:

- dist: Distribution type ("Exponential" or "Weibull")
- parameter_mode: Either "Fixed" or "Distribution"
- fixed_type: If "Fixed", specify as "Parameters" or "Landmark"
- lambda, gamma: Scale and shape parameters
- t1, t2: Landmark times
- surv_t1, surv_t2: Survival probabilities at landmarks
- t1_Beta_a, t1_Beta_b, diff_Beta_a, diff_Beta_b: Beta prior parameters

effect_model A named list specifying beliefs about the treatment effect:

- delay_SHELF, HR_SHELF: SHELF objects encoding beliefs
- delay_dist, HR_dist: Distribution types ("hist" by default)
- P_S: Probability that survival curves separate
- P_DTE: Probability of delayed separation, conditional on separation

recruitment_model A named list specifying the recruitment process:

- method: "power" or "PWC"
- period, power: Parameters for power model
- rate, duration: Comma-separated strings for PWC model

IA_model A named list specifying the censoring mechanism for the future data:

- events: Number of events which is 100% information fraction
- IF: The information fraction at which to censor and calculate BPP

analysis_model A named list specifying the final analysis and decision rule:

- method: e.g. "LRT", "WLRT", or "MW".
- alpha: one-sided type I error level.
- alternative_hypothesis: direction of the alternative (e.g. "one.sided").
- rho, gamma, t_star, s_star: additional parameters for WLRT or MW (if applicable).

n_sims Number of data sets to simulate (default is 100).

Value

A vector of length n_sims corresponding to the value of BPP for each simulated trial

Examples

```
#' set.seed(123)
control_model = list(dist = "Exponential",
                     parameter_mode = "Distribution",
                     t1 = 12,
                     t1_Beta_a = 20,
                     t1_Beta_b = 32)

effect_model = list(delay_SHELF = SHELF::fitdist(c(5.5, 6, 6.5),
                    probs = c(0.25, 0.5, 0.75), lower = 0, upper = 12),
                    delay_dist = "gamma",
                    HR_SHELF = SHELF::fitdist(c(0.5, 0.6, 0.7),
                    probs = c(0.25, 0.5, 0.75), lower = 0, upper = 1),
                    HR_dist = "gamma",
                    P_S = 1,
                    P_DTE = 0)

recruitment_model <- list(method = "power", period = 12, power = 1)

IA_model = list(events = 20, IF = 0.5)

analysis_model = list(method = "LRT",
                      alpha = 0.025,
                      alternative_hypothesis = "one.sided")

timing <- calibrate_BPP_timing(n_c = 15, n_t = 15,
                     control_model = control_model,
                     effect_model = effect_model,
                     recruitment_model = recruitment_model,
                     IA_model = IA_model,
                     analysis_model = analysis_model,
                     n_sims = 2)
```

---

cens_data *Censor a survival dataset*

Description

Applies administrative censoring to a survival dataset using one of three methods: fixed time, fixed number of events, or fixed information fraction. The input data must contain columns for pseudo survival time, recruitment time, and observed time.

Usage

```
cens_data(
  data,
  cens_method = "Time",
```

```
  cens_time = NULL,
  cens_IF = NULL,
  cens_events = NULL
)
```

Arguments

data	A dataframe containing uncensored survival data with columns: `pseudo_time`, `rec_time`, and `time`
cens_method	Censoring method: `"Time"` (default), `"Events"`, or `"IF"`
cens_time	Time point for censoring (required if `cens_method` = `"Time"`)
cens_IF	Information fraction for censoring (required if `cens_method` = `"IF"`)
cens_events	Number of events for censoring (required if `cens_method` = `"Events"`)

Value

A list containing:

data Censored dataframe with updated `status` and filtered rows

cens_events Number of events used for censoring (if applicable)

cens_time Time point used for censoring

sample_size Number of patients remaining after censoring

Examples

```
set.seed(123)
df <- data.frame(
  pseudo_time = rexp(20, rate = 0.1),
  rec_time = runif(20, 0, 12),
  time = rexp(20, rate = 0.1)
)
censored <- cens_data(df, cens_method = "Time", cens_time = 10)
str(censored)
```

INTEREST *INTEREST data set*

Description

A reconstructed survival data set for the INTEREST clinical trial

Usage

```
INTEREST
```

Format

A data frame with 710 rows and 2 variables:

Survival time Survival Time (in months)

Status Event indicator (0=Alive, 1=Dead)

Source

Reconstructed survival data set from the following publication: https://www.sciencedirect.com/science/article/pii/S01406736

MCMC_sample *MCMC_sample*

Description

An MCMC sample for the example given in Salsbury et al (2024)

Usage

```
MCMC_sample
```

Format

A data frame with 100000 rows and 1 variables:

x Sample from the MAP prior

Source

A MCMC sample for the control group for the example given in https://onlinelibrary.wiley.com/doi/full/10.1002/sim.10136. Three historical data seta are used to generate a Meta-Analytic-Predictive Prior distribution

---

REVEL *REVEL data set*

Description

A reconstructed survival data set for the REVEL clinical trial

Usage

```
REVEL
```

Format

A data frame with 625 rows and 2 variables:

Survival time Survival Time (in months)

Status Event indicator (0=Alive, 1=Dead)

Source

Reconstructed survival data set from the following publication: https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(14)60845-X/fulltext

---

sim_dte — *Simulates survival times for a delayed treatment effect (DTE) scenario, where the treatment group experiences a delayed onset of benefit. Control and treatment groups are generated under exponential or Weibull distributions.*

Description

Simulates survival times for a delayed treatment effect (DTE) scenario, where the treatment group experiences a delayed onset of benefit. Control and treatment groups are generated under exponential or Weibull distributions.

Usage

```
sim_dte(
  n_c,
  n_t,
  lambda_c,
  delay_time,
  post_delay_HR,
  dist = "Exponential",
  gamma_c = NULL
)
```

Arguments

`n_c`	The number of patients in the control group
`n_t`	The number of patients in the treatment group
`lambda_c`	The baseline hazard rate for the control group
`delay_time`	The length of delay before treatment effect begins
`post_delay_HR`	The hazard ratio after the delay period
`dist`	The distribution for the control group; must be one of "Exponential" (default) or "Weibull"
`gamma_c`	The shape parameter for the Weibull distribution (only used if `dist = "Weibull"`)

Value

A data frame with two columns:

`time`	Simulated survival times
`group`	Group assignment: "Control" or "Treatment"

Class: `data.frame`

Examples

```
set.seed(123)
sim_data <- sim_dte(n_c = 10, n_t = 10, lambda_c = 0.1,
                    delay_time = 6, post_delay_HR = 0.6)
head(sim_data)
```

`survival_test` *Calculate statistical significance on a survival dataset*

Description

Performs a survival analysis using either the standard log-rank test (LRT) or a weighted log-rank test (WLRT). The function estimates the hazard ratio and determines whether the result is statistically significant based on the specified alpha level and alternative hypothesis.

Usage

```
survival_test(
  data,
  analysis_method = "LRT",
  alternative = "one.sided",
  alpha = 0.05,
  rho = 0,
  gamma = 0,
  t_star = NULL,
  s_star = NULL
)
```

Arguments

`data` A dataframe containing survival data. Must include columns for survival time, event status, and treatment group.

`analysis_method`
Method of analysis: `"LRT"` (default) for standard log-rank test, or `"WLRT"` for weighted log-rank test.

`alternative` String specifying the alternative hypothesis. Must be one of `"one.sided"` or `"two.sided"` (default).

alpha	Type I error threshold for significance testing.
rho	Rho parameter for the Fleming-Harrington weighted log-rank test.
gamma	Gamma parameter for the Fleming-Harrington weighted log-rank test.
t_star	Parameter t^* used in modestly weighted tests.
s_star	Parameter s^* used in modestly weighted tests.

Value

A list containing:

Signif Logical indicator of statistical significance based on the chosen test and alpha level.

observed_HR Estimated hazard ratio from a Cox proportional hazards model.

Examples

```
set.seed(123)
df <- data.frame(
  survival_time = rexp(40, rate = 0.1),
  status = rbinom(40, 1, 0.8),
  group = rep(c("Control", "Treatment"), each = 20)
)
result <- survival_test(df, analysis_method = "LRT", alpha = 0.05)
str(result)
```

---

update_priors *Update prior distributions using interim survival data*

Description

This function updates elicited priors (defined through SHELF objects and parametric prior distributions) using interim survival data under a delayed-effect, piecewise-exponential model for the treatment arm and an exponential or Weibull model for the control arm.

Usage

```
update_priors(
  data,
  control_model,
  effect_model,
  n.chains = 2,
  n_burnin = 500,
  n_samples = 1000
)
```

Arguments

data — A data frame containing interim survival data with columns:

- survival_time Observed time from randomisation to event/censoring.
- status Event indicator (1 = event, 0 = censored).
- group Group identifier (e.g., "Control", "Treatment").

control_model — A named list specifying the control arm survival distribution:

- dist: Distribution type ("Exponential" or "Weibull")
- parameter_mode: Either "Fixed" or "Distribution"
- fixed_type: If "Fixed", specify as "Parameters" or "Landmark"
- lambda, gamma: Scale and shape parameters
- t1, t2: Landmark times
- surv_t1, surv_t2: Survival probabilities at landmarks
- t1_Beta_a, t1_Beta_b, diff_Beta_a, diff_Beta_b: Beta prior parameters

effect_model — A named list specifying beliefs about the treatment effect:

- delay_SHELF, HR_SHELF: SHELF objects encoding beliefs
- delay_dist, HR_dist: Distribution types ("hist" by default)
- P_S: Probability that survival curves separate
- P_DTE: Probability of delayed separation, conditional on separation

n.chains — Number of MCMC chains to run (default is 2)

n_burnin — Number of burn-in samples for the MCMC chain(s) (default is 500)

n_samples — Number of posterior samples to generate (default is 1000)

Value

A data frame containing Monte Carlo samples from the updated (posterior) distribution of the model parameters. Columns normally include:

- lambda_c Posterior samples for the control hazard parameter.
- delay_time Posterior samples for the delay/changepoint time T.
- HR Posterior samples for the post-delay hazard ratio.
- gamma_c (only if control_distribution = "Weibull") Posterior samples for the Weibull shape parameter.

Priors for lambda_c, T, and HR are constructed from elicited distributions using the SHELF framework, then updated through sampling-based posterior inference.

Examples

```
set.seed(123)
interim_data = data.frame(survival_time = runif(10, min = 0, max = 10),
status = rbinom(10, size = 1, prob = 0.5),
group = c(rep("Control", 5), rep("Treatment", 5)))
control_model = list(dist = "Exponential",
```

```
                        parameter_mode = "Distribution",
                        t1 = 12,
                        t1_Beta_a = 20,
                        t1_Beta_b = 32)

effect_model = list(delay_SHELF = SHELF::fitdist(c(5.5, 6, 6.5),
                    probs = c(0.25, 0.5, 0.75), lower = 0, upper = 12),
                    delay_dist = "gamma",
                    HR_SHELF = SHELF::fitdist(c(0.5, 0.6, 0.7),
                    probs = c(0.25, 0.5, 0.75), lower = 0, upper = 1),
                    HR_dist = "gamma",
                    P_S = 1,
                    P_DTE = 0)

posterior_df <- update_priors(
  data = interim_data,
  control_model = control_model,
  effect_model = effect_model,
  n_samples = 10)
```

`ZODIAC` *ZODIAC data set*

Description

A reconstructed survival data set for the ZODIAC clinical trial

Usage

```
ZODIAC
```

Format

A data frame with 697 rows and 2 variables:

Survival time Survival Time (in months)

Status Event indicator (0=Alive, 1=Dead)

Source

Reconstructed survival data set from the following publication: https://www.sciencedirect.com/science/article/abs/pii/S14702

Index