

Package ‘DTEAssurance’

October 24, 2025

Type Package

Title Assurance Methods for Clinical Trials with a Delayed Treatment Effect

Version 1.0.1

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](https://doi.org/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](https://doi.org/10.48550/arXiv.2509.07602)>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

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

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.0)

Date 2025-09-25

URL <https://jamesalsbury.github.io/DTEAssurance/>

NeedsCompilation no

Author James Salsbury [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-2584-3640>>)

Maintainer James Salsbury <jsalsbury1@sheffield.ac.uk>

Repository CRAN

Date/Publication 2025-10-24 14:30:02 UTC

Contents

add_recruitment_time	2
assurance_GSD_shiny_app	3
assurance_shiny_app	4
calc_dte_assurance	4
calc_dte_assurance_interim	6
cens_data	8
INTEREST	9
MCMC_sample	10
REVEL	10
sim_dte	11
survival_test	12
ZODIAC	13
Index	14

add_recruitment_time	<i>Add recruitment time to a survival dataset</i>
----------------------	---

Description

Simulates recruitment timing for each subject 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 subject

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_GSD_shiny_app

Launch the 'shiny' GSD 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_GSD_shiny_app()
```

Value

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

Examples

```
if (interactive()) {
  # Launch the interactive app in an R session
  assurance_GSD_shiny_app()
}
```

assurance_shiny_app	<i>Launch the 'shiny' Assurance app</i>
---------------------	---

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

```
if (interactive()) {  
  # Launch the interactive app in an R session  
  assurance_shiny_app()  
}
```

calc_dte_assurance	<i>Calculate Assurance for a Trial with a Delayed Treatment Effect</i>
--------------------	--

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

<code>n_c</code>	Vector of control group sample sizes
<code>n_t</code>	Vector of treatment group sample sizes
<code>control_model</code>	A named list specifying the control arm survival distribution: • <code>dist</code>: Distribution type ("Exponential" or "Weibull") • <code>parameter_mode</code>: Either "Fixed" or "Distribution" • <code>fixed_type</code>: If "Fixed", specify as "Parameters" or "Landmark" • <code>lambda</code>, <code>gamma</code>: Scale and shape parameters • <code>t1</code>, <code>t2</code>: Landmark times • <code>surv_t1</code>, <code>surv_t2</code>: Survival probabilities at landmarks • <code>t1_Beta_a</code>, <code>t1_Beta_b</code>, <code>diff_Beta_a</code>, <code>diff_Beta_b</code>: Beta prior parameters
<code>effect_model</code>	A named list specifying beliefs about the treatment effect: • <code>delay_SHELF</code>, <code>HR_SHELF</code>: SHELF objects encoding beliefs • <code>delay_dist</code>, <code>HR_dist</code>: Distribution types ("hist" by default) • <code>P_S</code>: Probability that survival curves separate • <code>P_DTE</code>: Probability of delayed separation, conditional on separation
<code>censoring_model</code>	A named list specifying the censoring mechanism: • <code>method</code>: "Time", "Events", or "IF" • <code>time</code>, <code>events</code>, <code>IF</code>: Parameters for each method
<code>recruitment_model</code>	A named list specifying the recruitment process: • <code>method</code>: "power" or "PWC" • <code>period</code>, <code>power</code>: Parameters for power model • <code>rate</code>, <code>duration</code>: Comma-separated strings for PWC model
<code>analysis_model</code>	A named list specifying the statistical test and decision rule: • <code>method</code>: "LRT", "WLRT", or "MW" • <code>alpha</code>, <code>alternative_hypothesis</code>: Type I error and hypothesis direction • <code>rho</code>, <code>gamma</code>, <code>t_star</code>, <code>s_star</code>: Parameters for WLRT or MW • <code>success_threshold_HR</code>: Optional threshold for declaring success
<code>n_sims</code>	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

```
# Minimal example with placeholder inputs
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_interim

*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_interim(
  n_c,
  n_t,
  control_model,
  effect_model,
  recruitment_model,
  GSD_model,
  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 • beta_spending: Cumulative beta spending vector • IF_vec: Vector of information fractions
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

```
# Minimal example with placeholder inputs
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("0.01, 0.025"),
  beta_spending = c("0.05, 0.1"), IF_vec = c("0.5, 1"))
result <- calc_dte_assurance_interim(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)
```

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 subjects 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	<i>MCMC_sample</i>
-------------	--------------------

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 sets are used to generate a Meta-Analytic-Predictive Prior distribution

REVEL	<i>REVEL data set</i>
-------	-----------------------

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](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(14)60845-X/fulltext)

sim_dte	<i>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.</i>
---------	---

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)
```

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

* datasets

INTEREST, [9](#)

MCMC_sample, [10](#)

REVEL, [10](#)

ZODIAC, [13](#)

add_recruitment_time, [2](#)

assurance_GSD_shiny_app, [3](#)

assurance_shiny_app, [4](#)

calc_dte_assurance, [4](#)

calc_dte_assurance_interim, [6](#)

cens_data, [8](#)

INTEREST, [9](#)

MCMC_sample, [10](#)

REVEL, [10](#)

sim_dte, [11](#)

survival_test, [12](#)

ZODIAC, [13](#)