Title  Causal Effects in Principal Strata Defined by Antidrug Antibodies

Version  0.0.2

Description  Bayesian models to estimate causal effects of biological treatments on time-to-event endpoints in clinical trials with principal strata defined by the occurrence of antidrug antibodies. The methodology is based on Frangakis and Rubin (2002) \(<doi:10.1111/j.0006-341x.2002.00021.x>\) and Imbens and Rubin (1997) \(<doi:10.1214/aos/1034276631>\), and intended to be applied to a specific time-to-event setting.

License  GPL (>= 3)

Encoding  UTF-8

URL  https://github.com/Boehringer-Ingelheim/BPrinStratTTE

BugReports  https://github.com/Boehringer-Ingelheim/BPrinStratTTE/issues

RoxygenNote  7.2.3

Imports  dplyr, furrr, magrittr, methods, purrr, Rcpp (>= 0.12.0), rstan (>= 2.18.1), stats, stringr, tibble

Biarch  true

Depends  R (>= 3.4.0)

LinkingTo  BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements  GNU make, C++17

Suggests  spelling

Language  en-US

NeedsCompilation  yes

Author  Christian Stock [aut, cre] (<https://orcid.org/0000-0002-3493-3234>), Emma Torrini [aut], Boehringer Ingelheim Pharma GmbH & Co. KG [cph, fnl]

Maintainer  Christian Stock <christian.stock@boehringer-ingelheim.com>
Description


References

fit_mult_exp_covar

Fit multiple models to data from two-arm trials with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event

Description

Fit multiple models to data from two-arm trials with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event

Usage

fit_mult_exp_covar(dat_mult_trials, params, seed = 23)

Arguments

dat_mult_trials
  List generated by sim_dat_mult_trials_exp_covar.
params
  List of model parameters as supplied to fit_single_exp_covar.
seed
  Numeric value, seed for reproducibility.

Value

A list of objects generated by fit_single_exp_covar.

See Also

sim_dat_mult_trials_exp_covar(), fit_single_exp_covar(), fit_mult_exp_nocovar()

Examples

d_params_covar <- list(
  n = 1000,
  nt = 500,
  prob_X1 = 0.4,
  prob_ice_X1 = 0.5,
  prob_ice_X0 = 0.2,
  fu_max = 48*7,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
dat_mult_trials <- sim_dat_mult_trials_exp_covar(
  n_iter = 2,
  params = d_params_covar
)
m_params_covar <- list(
  tg = 48,
fit_mult_exp_nocovar

Fit multiple models to data from two-arm trials with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Description

Fit multiple models to data from two-arm trials with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Usage

fit_mult_exp_nocovar(dat_mult_trials, params, seed = 23)

Arguments

dat_mult_trials
  List generated by sim_dat_mult_trials_exp_nocovar.

params
  List of model parameters as supplied to fit_single_exp_nocovar.

seed
  Numeric value, seed for reproducibility.
Value

A list of objects generated by `fit_single_exp_nocovar`.

See Also

`sim_dat_mult_trials_exp_nocovar()`, `fit_single_exp_nocovar()`, `fit_mult_exp_covar()`

Examples

d_params_nocovar <- list(
  n = 500L,
  nt = 250L,
  prob_ice = 0.5,
  fu_max = 336L,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
dat_mult_trials <- sim_dat_mult_trials_exp_nocovar(
  n_iter = 2,
  params = d_params_nocovar
)
m_params_nocovar <- list(
  tg = 48L,
  prior_piT = c(0.5, 0.5),
  prior_0N = c(1.5, 5),
  prior_1N = c(1.5, 5),
  prior_0T = c(1.5, 5),
  prior_1T = c(1.5, 5),
  t_grid = seq(7, 7 * 48, 7) / 30,
  chains = 2L,
  n_iter = 3000L,
  warmup = 1500L,
  cores = 2L,
  open_progress = FALSE,
  show_messages = TRUE
)

fit_multiple <- fit_mult_exp_nocovar(
  dat_mult_trials = dat_mult_trials,
  params = m_params_nocovar,
  seed = 12
)
lapply(fit_multiple, dim)
head(fit_multiple[[1]])
**fit_single_exp_covar**

*Fit single model to data from a two-arm trial with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event*

**Description**

Fit single model to data from a two-arm trial with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event

**Usage**

```r
fit_single_exp_covar(data, params, summarize_fit = TRUE)
```

**Arguments**

- **data**
  - Data frame of a structure as generated by `sim_dat_one_trial_exp_covar()`.

- **params**
  - List, containing model parameters:
    -  `tg` Positive integer value, number of intervals to calculate restricted mean survival time using the trapezoidal rule.
    -  `p` Positive integer value, number of predictors of the intercurrent event of interest (i.e. the event that determines the principal stratum membership).
    -  `prior_delta` px2 matrix of positive numerical values, containing normal priors (mean and standard deviation) of the model parameter delta.
    -  `prior_0N` Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_0N.
    -  `prior_1N` Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_1N.
    -  `prior_0T` Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_0T.
    -  `prior_1T` Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_1T.
    -  `t_grid` Numeric vector of length `tg`, containing time points defining the time grid (in months) to calculate restricted mean survival time using the trapezoidal rule.
    -  `chains` Positive integer value, specifying the number of Markov chains.
    -  `n_iter` Positive integer value, specifying the number of iterations for each chain (including warmup).
    -  `warmup` Positive integer value, specifying the number of warmup (aka burnin) iterations per chain.
    -  `cores` Positive integer value, specifying the number of cores to use when executing the chains in parallel.
    -  `open_progress` Logical value, indicating whether the progress of the chains will be redirected to a file that is automatically opened for inspection.
• show_messages Logical value, indicating whether to print the summary of informational messages.

summarize_fit Logical, if TRUE (default), the output is restricted to a summary of results on key parameters over all chains, if FALSE, the complete stanfit object is returned.

Details
The data supplied as params are used either as priors (prior_delta, prior_0N, prior_1N, prior_1T), to inform the model setup (tg, p, t_grid), or as parameters to rstan::sampling() which is invoked internally (chains, n_iter, warmup, cores, open_progress, show_messages).

Value
tibble() containing a summary of results on key parameters, or a stanfit object (S4 class), depending on summarize_fit.

See Also
fit_single_exp_nocovar() and rstan::sampling()

Examples
d_params_covar <- list(
n = 1000,
nt = 500,
prob_X1 = 0.4,
prob_ice_X1 = 0.5,
prob_ice_X0 = 0.2,
fu_max = 48*7,
T0T_rate = 0.2,
T0N_rate = 0.2,
T1T_rate = 0.15,
T1N_rate = 0.1
)
dat_single_trial <- sim_dat_one_trial_exp_covar(
n = d_params_covar[["n"]],
nt = d_params_covar[["nt"]],
prob_X1 = d_params_covar[["prob_X1"]],
prob_ice_X1 = d_params_covar[["prob_ice_X1"]],
prob_ice_X0 = d_params_covar[["prob_ice_X0"]],
fu_max = d_params_covar[["fu_max"]],
T0T_rate = d_params_covar[["T0T_rate"]],
T0N_rate = d_params_covar[["T0N_rate"]],
T1T_rate = d_params_covar[["T1T_rate"]],
T1N_rate = d_params_covar[["T1N_rate"]]
)
m_params_covar <- list(
tg = 48,
p = 2,
prior_delta = matrix(
  c(0, 5, 0, 5),
  nrow = 2, byrow = TRUE),
fit_single_exp_nocovar

Fit single model to data from a two-arm trial with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Usage

fit_single_exp_nocovar(data, params, summarize_fit = TRUE)

Arguments

data Data frame of a structure as generated by sim_dat_one_trial_exp_nocovar().

params List, containing model parameters:

- **tg** Positive integer value, number of intervals to calculate restricted mean survival time using the trapezoidal rule.
- **prior_piT** Numeric vector of length 2, containing parameters (alpha, beta) of the beta prior on pi, indicating the probability of belonging to the stratum of subjects developing the intercurrent event if given treatment.
- **prior_0N** Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_0N.
• prior_1N Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_1N.
• prior_0T Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_0T.
• prior_1T Numeric vector of length 2, containing parameters (alpha, beta) of the gamma prior on lambda_1T.
• t_grid Numeric vector of length \(tg\), containing time points defining the time grid (in months) to calculate restricted mean survival time using the trapezoidal rule.
• chains Positive integer value, specifying the number of Markov chains.
• n_iter Positive integer value, specifying the number of iterations for each chain (including warmup).
• warmup Positive integer value, specifying the number of warmup (aka burnin) iterations per chain.
• cores Positive integer value, specifying the number of cores to use when executing the chains in parallel.
• open_progress Logical value, indicating whether the progress of the chains will be redirected to a file that is automatically opened for inspection.
• show_messages Logical value, indicating whether to print the summary of informational messages.

\textbf{summarize_fit} Logical, if \texttt{TRUE} (default), the output is restricted to a summary of results on key parameters over all chains, if \texttt{FALSE}, the complete stanfit object is returned.

\textbf{Details}

The data supplied as \texttt{params} are used either as priors (\texttt{prior_delta, prior_0N, prior_1N, prior_1T}), to inform the model setup (\texttt{tg, p, t_grid}), or as parameters to \texttt{rstan::sampling()} which is invoked internally (\texttt{chains, n_iter, warmup, cores, open_progress, show_messages}).

\textbf{Value}

tibble() containing a summary of results on key parameters, or a \texttt{stanfit} object, depending on \texttt{summarize_fit}.

\textbf{See Also}

\texttt{fit_single_exp_covar()} and \texttt{rstan::sampling()}

\textbf{Examples}

\begin{verbatim}
d_params_nocovar <- list(n = 500L, nt = 250L, prob_ice = 0.5, fu_max = 336L, T0T_rate = 0.2, T0N_rate = 0.2, T1T_rate = 0.15, prior_1N = c(5, 2), prior_0T = c(5, 2), prior_1T = c(5, 2), t_grid = seq(0, 336, by = 1), chains = 4, n_iter = 1000, warmup = 500, cores = 2, open_progress = TRUE, show_messages = FALSE, summarize_fit = TRUE)
\end{verbatim}
T1N_rate = 0.1
)

dat_single_trial <- sim_dat_one_trial_exp_nocovar(  
n = d_params_nocovar["n"],  
nt = d_params_nocovar["nt"],  
prob_ice = d_params_nocovar["prob_ice"],  
fu_max = d_params_nocovar["fu_max"],  
T0T_rate = d_params_nocovar["T0T_rate"],  
T0N_rate = d_params_nocovar["T0N_rate"],  
T1T_rate = d_params_nocovar["T1T_rate"],  
T1N_rate = d_params_nocovar["T1N_rate"]
)
m_params_nocovar <- list(  
tg = 48L,  
prior_piT = c(0.5, 0.5),  
prior_0N = c(1.5, 5),  
prior_1N = c(1.5, 5),  
prior_0T = c(1.5, 5),  
prior_1T = c(1.5, 5),  
t_grid = seq(7, 7 * 48, 7) / 30,  
chains = 2L,  
n_iter = 3000L,  
warmup = 1500L,  
cores = 2L,  
open_progress = FALSE,  
show_messages = TRUE
)

fit_single <- fit_single_exp_nocovar(  
data = dat_single_trial,  
params = m_params_nocovar,  
summarize_fit = TRUE
)

print(fit_single)

---

**inv_logit**  
*Inverse logit function*

**Description**  
Inverse logit function

**Usage**  
`inv_logit(x)`

**Arguments**  
- **x**  
  Numeric value (usually a logarithm of odds).
Details

The inverse logit function is also known as logistic function.

Value

Numeric value on the interval \([0, 1]\), result of \(\log(\pi/(1-\pi))\).
Numeric value, result of \(\exp(x)/(1+\exp(x))\).

See Also

logit()

Examples

# probabilities
prob_ICE_base <- 0.3
prob_ICE_risk <- 0.6
# model coefficients
(beta1 <- logit(prob_ICE_base))
(beta2 <- logit(prob_ICE_risk) - logit(prob_ICE_base))
# linear predictor
logit(prob_ICE_base); (lin_pred1 <- beta1 + beta2*0)
logit(prob_ICE_risk); (lin_pred2 <- beta1 + beta2*1)
# inverse logit of linear predictor
(inv_logit(lin_pred1)) # prob for X1 = 0
(inv_logit(lin_pred2)) # prob for X1 = 1

logit

Logit function

Description

Logit function

Usage

logit(pi)

Arguments

pi

Numeric value on the interval \([0, 1]\) (usually a probability).

Value

Numeric value, result of \(\log(\pi/(1-\pi))\).

See Also

inv_logit()
Examples

```r
# probabilities
prob_ICE_base <- 0.3
prob_ICE_risk <- 0.6
# model coefficients
(beta1 <- logit(prob_ICE_base))
(beta2 <- logit(prob_ICE_risk) - logit(prob_ICE_base))
# linear predictor
logit(prob_ICE_base); (lin_pred1 <- beta1 + beta2*0)
logit(prob_ICE_risk); (lin_pred2 <- beta1 + beta2*1)
# inverse logit of linear predictor
(inv_logit(lin_pred1)) # prob for X1 = 0
(inv_logit(lin_pred2)) # prob for X1 = 1
```

ocs_exp_covar  

Determine operating characteristics of fits from two-arm trials with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event

Description

Determine operating characteristics of fits from two-arm trials with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event

Usage

```r
ocs_exp_covar(multiple_fits, d_params, m_params)
```

Arguments

- `multiple_fits` List of model fits from `fit_mult_exp_covar`.
- `d_params` List of data parameters as used in `sim_dat_one_trial_exp_covar`.
- `m_params` List of model parameters as used in `fit_single_exp_covar`.

Details

This function is used in `run_sim_exp_covar()`, the output of the two functions is the same.

Value

A list of length 3, containing objects call `ocs`, `d_params`, `m_params`, where `ocs` is a tibble containing averaged parameter estimates and operating characteristics, and `d_params` and `m_params` are the objects supplied to the function.

See Also

`ocs_exp_nocovar()` and `run_sim_exp_covar()`.


Examples

```r

d_params_covar <- list(
  n = 1000,
  nt = 500,
  prob_X1 = 0.4,
  prob_ice_X1 = 0.5,
  prob_ice_X0 = 0.2,
  fu_max = 48*7,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)

dat_mult_trials <- sim_dat_mult_trials_exp_covar(
  n_iter = 2,
  params = d_params_covar
)

m_params_covar <- list(
  tg = 48,
  p = 2,
  prior_delta = matrix(
    c(0, 5, 0, 5),
    nrow = 2, byrow = TRUE),
  prior_0N = c(1.5, 5),
  prior_1N = c(1.5, 5),
  prior_0T = c(1.5, 5),
  prior_1T = c(1.5, 5),
  t_grid = seq(7, 7 * 48, 7) / 30,
  chains = 2,
  n_iter = 3000,
  warmup = 1500,
  cores = 2,
  open_progress = FALSE,
  show_messages = TRUE
)

fit_multiple <- fit_mult_exp_covar(
  dat_mult_trials = dat_mult_trials,
  params = m_params_covar,
  seed = 12
)

list_ocs <- ocs_exp_covar(
  multiple_fits = fit_multiple,
  d_params = d_params_covar,
  m_params = m_params_covar
)

print(list_ocs)
```
Determine operating characteristics of fits from two-arm trials with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Usage
ocs_exp_nocovar(multiple_fits, d_params, m_params)

Arguments
multiple_fits  List of model fits from fit_mult_exp_nocovar.
d_params  List of data parameters as used in sim_dat_one_trial_exp_nocovar.
m_params  List of model parameters as used in fit_single_exp_nocovar.

Details
This function is used in run_sim_exp_nocovar(), the output of the two functions is the same.

Value
A list of length 3, containing objects call ocs, d_params, m_params, where ocs is a tibble containing averaged parameter estimates and operating characteristics, and d_params and m_params are the objects supplied to the function.

See Also
ocs_exp_covar() and run_sim_exp_nocovar().

Examples
d_params_nocovar <- list(
  n = 500L,
  nt = 250L,
  prob_ice = 0.5,
  fu_max = 336L,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
dat_mult_trials <- sim_dat_mult_trials_exp_nocovar(
  n_iter = 2,


### Description

Run simulation of two-arm trials with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event.

### Usage

```r
run_sim_exp_covar(n_iter, d_params, m_params, seed)
```

### Arguments

- **n_iter**: Positive integer value, number of trials to be simulated.
- **d_params**: List of data parameters as used in `sim_dat_one_trial_exp_nocovar`.
- **m_params**: List of model parameters as used in `fit_single_exp_nocovar`.
- **seed**: Numeric value, seed for reproducibility.
Value

A list of length 3, containing objects call ocs, d_params, m_params, where ocs is a tibble containing averaged parameter estimates and operating characteristics, and d_params and m_params are the objects supplied to the function.

See Also

run_sim_exp_nocovar()

Examples

d_params_covar <- list(
  n = 1000,
  nt = 500,
  prob_X1 = 0.4,
  prob_ice_X1 = 0.5,
  prob_ice_X0 = 0.2,
  fu_max = 48*7,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
m_params_covar <- list(
  tg = 48,
  p = 2,
  prior_delta = matrix(
    c(0, 5, 0, 5),
    nrow = 2, byrow = TRUE),
  prior_0N = c(1.5, 5),
  prior_1N = c(1.5, 5),
  prior_0T = c(1.5, 5),
  prior_1T = c(1.5, 5),
  t_grid = seq(7, 7 * 48, 7) / 30,
  chains = 2,
  n_iter = 3000,
  warmup = 1500,
  cores = 2,
  open_progress = FALSE,
  show_messages = TRUE
)

dat_ocs <- run_sim_exp_covar(
  n_iter = 3,
  d_params = d_params_covar,
  m_params = m_params_covar,
  seed = 12
)
print(dat_ocs)
run_sim_exp_nocovar

Run simulation of two-arm trials with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Description

Run simulation of two-arm trials with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Usage

run_sim_exp_nocovar(n_iter, d_params, m_params, seed)

Arguments

n_iter
  Positive integer value, number of trials to be simulated.

d_params
  List of data parameters as used in sim_dat_one_trial_exp_nocovar.

m_params
  List of model parameters as used in fit_single_exp_nocovar.

seed
  Numeric value, seed for reproducibility.

Value

A list of length 3, containing objects call ocs, d_params, m_params, where ocs is a tibble containing averaged parameter estimates and operating characteristics, and d_params and m_params are the objects supplied to the function.

See Also

run_sim_exp_covar()

Examples

d_params_nocovar <- list(
  n = 500L,
  nt = 250L,
  prob_ice = 0.5,
  fu_max = 336L,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)

m_params_nocovar <- list(
  tg = 48L,
  prior_piT = c(0.5, 0.5),
  prior_0N = c(1.5, 5),
  prior_1N = c(1.5, 5),
  prior_0T = c(1.5, 5),
)
prior_1T = c(1.5, 5),
t_grid = seq(7, 7 * 48, 7) / 30,
chains = 2L,
n_iter = 3000L,
warmup = 1500L,
cores = 2L,
open_progress = FALSE,
show_messages = TRUE
)

dat_ocs <- run_sim_exp_nocovar(
  n_iter = 3,
  d_params = d_params_nocovar,
  m_params = m_params_nocovar,
  seed = 12
)
print(dat_ocs)

---

**sim_dat_mult_trials_exp_covar**

*Simulate data from multiple two-arm trials with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event*

### Description

Simulate data from multiple two-arm trials with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event

### Usage

`sim_dat_mult_trials_exp_covar(n_iter, params)`

### Arguments

- **n_iter**: Positive integer value, number of trials to be simulated.
- **params**: List of data parameters as used in `sim_dat_one_trial_exp_covar`.

### Value

A list of length `n_iter`, containing objects of class `tibble()`, each containing one simulated trial dataset.

### See Also

`sim_dat_mult_trials_exp_nocovar()`
Examples

d_params_covar <- list(
  n = 1000,
  nt = 500,
  prob_X1 = 0.4,
  prob_ice_X1 = 0.5,
  prob_ice_X0 = 0.2,
  fu_max = 48*7,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
dat_mult_trials <- sim_dat_mult_trials_exp_covar(
  n_iter = 3,
  params = d_params_covar
)
lapply(dat_mult_trials, dim)
head(dat_mult_trials[[1]])

---

Simulate data from multiple two-arm trials with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Description

Simulate data from multiple two-arm trials with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event

Usage

sim_dat_mult_trials_exp_nocovar(n_iter, params)

Arguments

- `n_iter` Positive integer value, number of trials to be simulated.
- `params` List of data parameters as used in `sim_dat_one_trial_exp_nocovar`.

Value

A list of length `n_iter`, containing objects of class `tibble()`, each containing one simulated trial dataset.

See Also

`sim_dat_mult_trials_exp_covar()`
Examples

d_params_nocovar <- list(
  n = 500L,
  nt = 250L,
  prob_ice = 0.5,
  fu_max = 336L,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)

dat_mult_trials <- sim_dat_mult_trials_exp_nocovar(
  n_iter = 3,
  params = d_params_nocovar
)
lapply(dat_mult_trials, dim)
head(dat_mult_trials[[1]])

---

**sim_dat_one_trial_exp_covar**

*Simulate data from a single two-arm trial with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event*

---

**Description**

Simulate data from a single two-arm trial with an exponentially distributed time-to-event endpoint and one predictor of the intercurrent event

**Usage**

```r
sim_dat_one_trial_exp_covar(
  n,
  nt,
  prob_X1,
  prob_ice_X1,
  prob_ice_X0,
  fu_max,
  T0T_rate,
  T0N_rate,
  T1T_rate,
  T1N_rate
)
```
Arguments

- **n**: Positive integer value, number of subjects in the trial.
- **nt**: Positive integer value, number of treated subjects.
- **prob_X1**: Numeric value on the interval $(0, 1)$, probability of being at high risk of experiencing the intercurrent event of interest when treated (i.e. the event that determines the principal stratum membership).
- **prob_ice_X1**: Numeric value on the interval $(0, 1)$, probability of the intercurrent event of interest if treated and at high risk of the intercurrent event.
- **prob_ice_X0**: Numeric value on the interval $(0, 1)$, probability of the intercurrent event of interest if treated and not at high risk of the intercurrent event.
- **fu_max**: Positive integer value, maximum follow-up time in days (administrative censoring assumed afterwards).
- **T0T_rate**: Positive numeric value, monthly event rate in control subjects that would develop the intercurrent event if treated.
- **T0N_rate**: Positive numeric value, monthly event rate in control subjects that never develop the intercurrent event.
- **T1T_rate**: Positive numeric value, monthly event rate in treated subjects that develop the intercurrent event.
- **T1N_rate**: Positive numeric value, monthly event rate in treated subjects that never develop the intercurrent event.

Value

...

See Also

- `sim_dat_one_trial_exp_nocovar()`

Examples

```r
  d_params_covar <- list(
    n = 1000,
    nt = 500,
    prob_X1 = 0.4,
    prob_ice_X1 = 0.5,
    prob_ice_X0 = 0.2,
    fu_max = 48*7,
    T0T_rate = 0.2,
    T0N_rate = 0.2,
    T1T_rate = 0.15,
    T1N_rate = 0.1
  )
  dat_single_trial <- sim_dat_one_trial_exp_covar(
    n = d_params_covar["n"],
    nt = d_params_covar["nt"],
    prob_X1 = d_params_covar["prob_X1"],
  ```
prob_ice_X1 = d_params_covar["prob_ice_X1"],
prob_ice_X0 = d_params_covar["prob_ice_X0"],
fu_max = d_params_covar["fu_max"],
T0T_rate = d_params_covar["T0T_rate"],
T0N_rate = d_params_covar["T0N_rate"],
T1T_rate = d_params_covar["T1T_rate"],
T1N_rate = d_params_covar["T1N_rate"]
)
dim(dat_single_trial)
head(dat_single_trial)

---

**sim_dat_one_trial_exp_nocovar**

*Simulate data from a single two-arm trial with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event*

**Description**

Simulate data from a single two-arm trial with an exponentially distributed time-to-event endpoint and no predictor of the intercurrent event.

**Usage**

```r
sim_dat_one_trial_exp_nocovar(
  n,
  nt,
  prob_ice,
  fu_max,
  T0T_rate,
  T0N_rate,
  T1T_rate,
  T1N_rate
)
```

**Arguments**

- **n**  Positive integer value, number of subjects in the trial.
- **nt**  Positive integer value, number of treated subjects.
- **prob_ice**  Numeric value on the interval $(0,1)$, probability of the intercurrent event of interest (i.e. the event that determines the principal stratum membership).
- **fu_max**  Positive integer value, maximum follow-up time in days (administrative censoring assumed afterwards).
- **T0T_rate**  Positive numeric value, monthly event rate in control subjects that would develop the intercurrent event if treated.
true_vals_exp_covar 23

T0N_rate Positive numeric value, monthly event rate in control subjects that never develop the intercurrent event.
T1T_rate Positive numeric value, monthly event rate in treated subjects that develop the intercurrent event.
T1N_rate Positive numeric value, monthly event rate in treated subjects that never develop the intercurrent event.

Value
A tibble() containing the trial data for analysis.

See Also
sim_dat_one_trial_exp_covar()

Examples

d_params_nocovar <- list(
  n = 500L,
  nt = 250L,
  prob_ice = 0.5,
  fu_max = 336L,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
dat_single_trial <- sim_dat_one_trial_exp_nocovar(
  n = d_params_nocovar["n"],
  nt = d_params_nocovar["nt"],
  prob_ice = d_params_nocovar["prob_ice"],
  fu_max = d_params_nocovar["fu_max"],
  T0T_rate = d_params_nocovar["T0T_rate"],
  T0N_rate = d_params_nocovar["T0N_rate"],
  T1T_rate = d_params_nocovar["T1T_rate"],
  T1N_rate = d_params_nocovar["T1N_rate"]
)
dim(dat_single_trial)
head(dat_single_trial)

true_vals_exp_covar Adding true values to estimates for models with an exponential endpoint and consideration of predictors of the intercurrent event

Description
Adding true values to estimates for models with an exponential endpoint and consideration of predictors of the intercurrent event
true_vals_exp_covar

Usage

true_vals_exp_covar(x, d_params, m_params)

Arguments

x Model object as returned by fit_single_exp_covar().
d_params List of data parameters as used in fit_single_exp_covar().
m_params List of model parameters as used in fit_single_exp_covar().

Value

A summary table with parameter estimates, true values and differences.

See Also

true_vals_exp_nocovar()

Examples

d_params_covar <- list(
  n = 1000,
  nt = 500,
  prob_X1 = 0.4,
  prob_ice_X1 = 0.5,
  prob_ice_X0 = 0.2,
  fu_max = 48*7,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
dat_single_trial <- sim_dat_one_trial_exp_covar(
  n = d_params_covar["n"],
  nt = d_params_covar["nt"],
  prob_X1 = d_params_covar["prob_X1"],
  prob_ice_X1 = d_params_covar["prob_ice_X1"],
  prob_ice_X0 = d_params_covar["prob_ice_X0"],
  fu_max = d_params_covar["fu_max"],
  T0T_rate = d_params_covar["T0T_rate"],
  T0N_rate = d_params_covar["T0N_rate"],
  T1T_rate = d_params_covar["T1T_rate"],
  T1N_rate = d_params_covar["T1N_rate"]
)
m_params_covar <- list(
  tg = 48,
  p = 2,
  prior_delta = matrix(
    c(0, 5, 0, 5),
    nrow = 2, byrow = TRUE),
  prior_0N = c(1.5, 5),
  prior_1N = c(1.5, 5),

true_vals_exp_nocovar

```r
prior_0T = c(1.5, 5),
prior_1T = c(1.5, 5),
t_grid = seq(7, 7 * 48, 7) / 30,
chains = 2,
n_iter = 3000,
warmup = 1500,
cores = 2,
open_progress = FALSE,
show_messages = TRUE
)

fit_single <- fit_single_exp_covar(
data = dat_single_trial,
params = m_params_covar,
summarize_fit = TRUE
)
print(fit_single)
tab_obs_truth <- true_vals_exp_covar(
  x = fit_single,
d_params = d_params_covar,
m_params = m_params_covar
)
print(tab_obs_truth)
```

true_vals_exp_nocovar  Adding true values to estimates for models with an exponential endpoint and no consideration of predictors of the intercurrent event

Description

Adding true values to estimates for models with an exponential endpoint and no consideration of predictors of the intercurrent event

Usage

```r
true_vals_exp_nocovar(x, d_params, m_params)
```

Arguments

- `x`  Model object as returned by `fit_single_exp_nocovar()`.
- `d_params`  List of data parameters as used in `fit_single_exp_nocovar()`.
- `m_params`  List of model parameters as used in `fit_single_exp_nocovar()`.

Value

A summary table with parameter estimates, true values and differences.
See Also

`true_vals_exp_covar()`

Examples

d_params_nocovar <- list(
  n = 500L,
  nt = 250L,
  prob_ice = 0.5,
  fu_max = 336L,
  T0T_rate = 0.2,
  T0N_rate = 0.2,
  T1T_rate = 0.15,
  T1N_rate = 0.1
)
dat_single_trial <- sim_dat_one_trial_exp_nocovar(
  n = d_params_nocovar["n"],
  nt = d_params_nocovar["nt"],
  prob_ice = d_params_nocovar["prob_ice"],
  fu_max = d_params_nocovar["fu_max"],
  T0T_rate = d_params_nocovar["T0T_rate"],
  T0N_rate = d_params_nocovar["T0N_rate"],
  T1T_rate = d_params_nocovar["T1T_rate"],
  T1N_rate = d_params_nocovar["T1N_rate"]
)
m_params_nocovar <- list(
  tg = 48L,
  prior_piT = c(0.5, 0.5),
  prior_0N = c(1.5, 5),
  prior_1N = c(1.5, 5),
  prior_0T = c(1.5, 5),
  prior_1T = c(1.5, 5),
  t_grid = seq(7, 7 * 48, 7) / 30,
  chains = 2L,
  n_iter = 3000L,
  warmup = 1500L,
  cores = 2L,
  open_progress = FALSE,
  show_messages = TRUE
)

fit_single <- fit_single_exp_nocovar(
  data = dat_single_trial,
  params = m_params_nocovar,
  summarize_fit = TRUE
)
print(fit_single)
tab_obs_truth <- true_vals_exp_nocovar(
  x = fit_single,
  d_params = d_params_nocovar,
  m_params = m_params_nocovar
)
true_vals_exp_nocovar

print(tab_obs_truth)
## Index

BPrinStratTTE (BPrinStratTTE-package), 2
BPrinStratTTE-package, 2

fit_multi_exp_covar, 3
fit_multi_exp_covar(), 5
fit_multi_exp_nocovar, 4
fit_multi_exp_nocovar(), 3
fit_single_exp_covar, 6
fit_single_exp_covar(), 3, 9
fit_single_exp_nocovar, 8
fit_single_exp_nocovar(), 5, 7

inv_logit, 10
inv_logit(), 11

logit, 11
logit(), 11

ocs_exp_covar, 12
ocs_exp_covar(), 14
ocs_exp_nocovar, 14
ocs_exp_nocovar(), 12

rstan::sampling, 7, 9
run_sim_exp_covar, 15
run_sim_exp_covar(), 12, 17
run_sim_exp_nocovar, 17
run_sim_exp_nocovar(), 14, 16

sim_dat_multi_trials_exp_covar, 18
sim_dat_multi_trials_exp_covar(), 3, 19
sim_dat_multi_trials_exp_nocovar, 19
sim_dat_multi_trials_exp_nocovar(), 5, 18

sim_dat_one_trial_exp_covar, 20
sim_dat_one_trial_exp_covar(), 23
sim_dat_one_trial_exp_nocovar, 22
sim_dat_one_trial_exp_nocovar(), 21

true_vals_exp_covar, 23
true_vals_exp_covar(), 26
true_vals_exp_nocovar, 25
true_vals_exp_nocovar(), 24