# Package 'seamlesssim'

September 28, 2020

**Title** Simulates Seamless Phase I/II Oncology Trials

**Version** 0.0.0.9000

**Description** seamlesssim simulates complex seamless Phase I/II oncology trials as discussed in the article by Boonstra et al. (Arxiv, 2020). It allows clinical trialists to determine operating characteristics of trials that assess both toxicity and efficacy with a range of different design and analytic approaches. For more information, see Boonstra et al. (Arxiv, 2020) and the vignette.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Biarch** true

**Depends** R (>= 3.4.0)

**Imports** methods,
Rcpp (>= 0.12.0),
rstan (>= 2.18.1),
rstantools (>= 2.0.0),
dfcrm,
dplyr,
tidyr,
ggplot2,
tibble,
stats,
data.table,
sjmisc,
RColorBrewer,
binom,
forcats,
purrr,
cowplot,
knitr

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

**VignetteBuilder** knitr

**SystemRequirements** GNU make

**RoxygenNote** 7.0.2

# R topics documented:



---

`seamlesssim-package` *The 'seamlesssim' package.*

---

## Description

seamlesssim simulates complex seamless Phase I/II oncology trials as discussed in the article by Boonstra et al. (Arxiv, 2020). It allows clinical trialists to determine operating characteristics of trials that assess both toxicity and efficacy with a range of different design and analytic approaches. For more information, see Boonstra et al. (Arxiv, 2020) and the vignette.

## References

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2. https://mc-stan.org

PS Boonstra, TM Braun, and EC Chase, "A modular framework for seamless oncology trials." Arxiv, 2020.

PS Boonstra, DR Owen, and J Kang, "The isotonic horseshoe prior for modeling binary outcomes." Arxiv, 2020.

Ken Cheung (2019). dfcrm: Dose-Finding by the Continual Reassessment Method. R package version 0.2-2.1. https://CRAN.R-project.org/package=dfcrm

---

`bayesian_isotonic` *Function to fit a Bayesian isotonic regression*

---

## Description

This function fits a Bayesian isotonic regression: it calculates the posterior distribution of a set of probabilities that are order constrained but each a priori marginally distributed as beta random variables.

## Usage

```
bayesian_isotonic(
  data_grouped = NULL,
  stan_args = list(local_dof_stan = 1, global_dof_stan = 1, alpha_scale_stan = 1),
  sample_from_prior_only = F,
  conf_level = 0.5,
  conf_level_direction = "both",
  verbose = F,
  n_mc_warmup = 1000,
  n_mc_samps = 2000,
  mc_chains = 4,
  mc_thin = 1,
  mc_stepsize = 0.1,
  mc_adapt_delta = 0.8,
  mc_max_treedepth = 15,
  ntries = 2,
  return_as_stan_object = F,
  tol = .Machine$double.eps^0.5,
  seed = 1
)
```

## Arguments

| | |
|---|---|
| data_grouped | A data.frame or tibble that contain columns x (used as category labels), y (the number of successes or events), and n (the number of trials). |
| stan_args | A named list of arguments to pass to the stan function. It should include the named elements:<br>**local_dof_stan** This has default value 1.<br>**global_dof_stan** This has default value 1.<br>**alpha_scale_stan** This has default value 1.<br>For more information on these arguments, please see rstan documentation. |
| sample_from_prior_only | A logical value. If TRUE, then the provided values of 'x' and 'n' will be ignored and draws will only be sampled from the prior distribution. Defaults to FALSE. |
| conf_level | Numeric in (0, 1). This level credible interval will be returned based upon the empirical quantiles. The default is 0.5. |
| conf_level_direction | This is a string equal to 'both', 'upper', or 'lower' indicating the desired direction of the credible interval. The default is 'both'. |
| verbose | A logical value. If TRUE, then all posterior draws will be returned by the function. Defaults to FALSE. |
| n_mc_warmup | A positive integer giving the number of desired warmup runs; the default is 1000. |
| n_mc_samps | A positive integer giving the number of additional samples to run after warmup is completed; the default is 2000. |
| mc_chains | A positive integer indicating the number of chains to run in parallel, which will multiply the final number of samples; the default is 4. |
| mc_thin | A positive integer indicating the number of iterations to thin by (increasing thinning will decrease the final number of samples); the default is 1. |

`mc_stepsize` A numeric value between 0 and 1 that is passed to control in the call to stan() as the stepsize argument; the default is 0.1.

`mc_adapt_delta` A numeric value between 0 and 1 that is passed to control in the call to stan() as the adapt_delta argument; the default is 0.8.

`mc_max_treedepth` A positive integer passed to control in the call to stan() as the max_treedepth argument; the default is 15.

`ntries` A positive integer. The stan algorithm throws warnings about divergent transitions, which are indicative of an inability to fully explore the posterior space. Sometimes this number can be extremely large, which suggests that the fitted model needs to be reparametrized. However, in this case, divergent transitions seem to be sporadic. ntries indicates how many reruns of the algorithm should be tried when > 0 divergent transitions are encountered. The run with the fewest such transitions is kept. The default is 2.

`return_as_stan_object` A logical value. If TRUE, then the function returns an object of class 'stanfit'. If FALSE, then a summary of results will be returned. Defaults to FALSE.

`tol` A small positive number (double). The default is the square root of the machine precision.

`seed` A positive integer used to set the random starting point for Stan sampling.

## Value

If return_as_stan_object = TRUE, then an object of class stanfit is returned. This is useful for the initial compilation of the stan model. Otherwise, the function returns the following named list containing the arguments:

**data_grouped** The provided argument of the same name but with more columns added that provide summaries of the efficacy probability for each row.

**conf_level, conf_level_direction** Arguments provided by the user for the confidence interval.

**stan_args** The list of arguments that were passed to the stan function.

**accepted_divergences** The number of divergent transitions from the model.

**max_divergences** The maximum observed number of divergent transitions from the ntries number of model fits that were attempted.

**rhat** The largest value of the Gelman-Rubin diagnostic across all parameters.

**number_nan** The number of draws that were 0/0, e.g. due to underflow.

**all_draws** NA if verbose == FALSE, otherwise all_draws is a matrix of draws from the posterior distribution, which may be large.

**chain_run_times_secs, total_run_time_secs** Length of time (seconds) for chain runs and total runs.

## References

PS Boonstra, DR Owen, and J Kang, "The isotonic horseshoe prior for modeling binary outcomes." Arxiv, 2020.

## calc_new_skeleton *Function to update a CRM skeleton in a two-stage CRM.*

### Description

This function updates an initial crm skeleton when using the empiric (power) model, after some data have been gathered on the dose-toxicity curve. This code is currently not used, but is included for posterity.

### Usage

```
calc_new_skeleton(old_skeleton, tox, level, offset_seq = NULL)
```

### Arguments

| | |
|---|---|
| old_skeleton | A numeric vector with entries between 0 and 1; the initial skeleton. |
| tox | A vector with binary entries 0,1 representing the outcomes of the patients; should be the same length as the number of observations collected. |
| level | A positive integer vector of labels with length equal to that of tox indicating the dose level that each observation was assigned to. Every element of this vector should, in addition to being positive, be no larger than the length of skeleton, i.e. the number of dose levels. |
| offset_seq | A positive numeric vector of arbitrary length that should probably have some values above and below 1. The old_skeleton corresponds to a value of 1, and including offset_seq helps to see if a better fit is obtained by scaling old_skeleton up or down. |

### Value

A named list containing elements new_skeleton (the updated skeleton) and new_scale (the suggested revised prior scale to put on the single parameter beta in the power model).

## print.philsim *S3 print method for titesim_phil*

### Description

print.philsim prints the results from titesim_phil, the simulator function for a tite-crm trial.

### Usage

```
## S3 method for class 'philsim'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of type titesim_phil, outputted from titesim_phil. |
| ... | List of additional arguments. Currently the following are implemented: |
| dgt | Number of digits for output rounding. Default value is 3. |
| patient.detail | Should patient-level information be provided, or only summary data? Default is patient-level, indicated by TRUE. For only summary data, use FALSE. |

### Value

Printed output from titesim_phil.

---

## sim_3pl3 *Function to simulate a 3+3 trial design*

---

### Description

This function is an efficient simulator of the 3+3 design. It is intended to be called from twostage_simulator rather than by the user directly.

### Usage

```
sim_3pl3(
  n_sim,
  true_tox_curve,
  stage_label = 1,
  sim_specific_start_id = NULL,
  sim_specific_dose_start = NULL,
  seed = sample(.Machine$integer.max, 1)
)
```

### Arguments

| | |
|---|---|
| n_sim | How many simulated trials to conduct? (positive integer) |
| true_tox_curve | A positive numeric vector that contains the true generating dose-toxicity curve to simulate the data from. This also gives the number of dose levels. |
| stage_label | A numeric value that can be arbitrary but is intended to take on integer values equal to either 1 or 2, corresponding to the stage of the trial. |
| sim_specific_start_id | A positive integer vector that contains the starting subject id for each simulated trial. If provided, it must be as long as 'n_sim.' If left blank, the default starting patient is patient 1. |
| sim_specific_dose_start | A positive integer vector that contains the starting dose level for each simulated trial. If provided, it must be as long as 'n_sim' and take on values between 0 (indicating that no patients should be enrolled) to length(true_tox_curve). If left blank, the default starting dose is dose 1. |
| seed | A positive integer seed for use prior to starting the simulations. |

## Value

A named list with entries:

**all_results** A matrix giving the individual patient outcomes from all simulated trials.

**estMTD** A vector as long as the number of simulated trials giving an integer value corresponding to the estimated MTD from each trial; a value of 0 indicates that all dose levels were estimated to be unsafe.

**enrollment** A vector as long as the number of simulated trials giving the total enrollment for that trial.

**seed** The seed that was used by the function.

---

`sim_empiric_dec` *A function to simulate a dose expansion cohort trial design*

---

## Description

A general purpose simulator for dose expansion cohorts (DECs). It is intended to be called from twostage_simulator rather than by the user directly.

## Usage

```
sim_empiric_dec(
  n_sim,
  true_tox_curve,
  stage_label = 1,
  sim_specific_start_id = NULL,
  sim_specific_dose_start = NULL,
  max_n_per_dec,
  module_rule = "local",
  thresh_decrease = 1/3,
  first_patient_look = 0,
  seed = sample(.Machine$integer.max, 1)
)
```

## Arguments

| | |
|---|---|
| `n_sim` | How many simulated trials to conduct? (positive integer) |
| `true_tox_curve` | A positive numeric vector that contains the true generating dose-toxicity curve to simulate the data from. This also gives the number of dose levels. |
| `stage_label` | A numeric value that can be arbitrary but is intended to take on integer values equal to either 1 or 2, corresponding to the stage of the trial. |
| `sim_specific_start_id` | A positive integer vector that contains the starting subject id for each simulated trial. If provided, it must be as long as n_sim. |
| `sim_specific_dose_start` | A positive integer vector that contains the starting dose level for each simulated trial. If provided, it must be as long as n_sim and take on values between 0 (indicating that no patients should be enrolled) to length(true_tox_curve). |

`max_n_per_dec` A positive integer giving the maximum enrollment for each DEC, if no stopping for toxicity occurs.

`module_rule` Currently the only valid choice for this is the string "local", corresponding to a local stopping rule: if the empiric proportion of DLTs at the current dose level ever exceeds thresh_decrease, then de-escalate.

`thresh_decrease` A numeric value between 0 and 1. This is the maximum tolerance for the observed proportion of DLTs at any given dose level. If this threshold is ever exceeded, the DEC will de-escalate if possible or stop the trial entirely if not.

`first_patient_look` An integer greater than or equal to 0 and less than or equal to max_n_per_dec. At a given dose level, at what point should the module_rule start taking action? This is an ad-hoc way to prevent a scenario such as "The first patient experienced a DLT, therefore because all patients at this dose level have experienced a DLT, we should de-escalate."

`seed` A positive integer seed for use prior to starting the simulations.

## Value

A named list with entries:

**all_results** A matrix giving the individual patient outcomes from all simulated trials.

**estMTD** A vector as long as the number of simulated trials giving an integer value corresponding to the estimated MTD from each trial; a value of 0 indicates that all dose levels were estimated to be unsafe.

**enrollment** A vector as long as the number of simulated trials giving the total enrollment for that trial.

**seed** The seed that was used by the function.

---

# `sim_twostage_crm` *Function to simulate a two-stage CRM*

---

## Description

This is a general purpose simulator for the CRM as used by the function twostage_simulator. If desired, users can simulate a two-stage CRM (two rounds of toxicity assignment), rather than a one-stage CRM.

## Usage

```
sim_twostage_crm(
  n_sim,
  titecrm_args,
  second_stage_start_after = Inf,
  first_stage_label = 1,
  sim_specific_start_id = NULL,
  sim_specific_prior = NULL,
  sim_specific_x0 = NULL,
  sim_specific_scale = NULL,
  seed = sample(.Machine$integer.max, 1)
)
```

## Arguments

| | |
|---|---|
| n_sim | How many simulated trials to conduct? (positive integer) |
| titecrm_args | This is a named list providing all of the arguments that the function titesim_phil expects. See titesim_phil documentation for more information on these required components. If the arguments prior, x0, and scale vary between simulations, then these arguments should be specified separately using sim_specific_prior, sim_specific_x0, and sim_specific_scale, respectively. |
| second_stage_start_after | If simulating a two-stage CRM, this positive integer indicates after how many patients should we switco the second stage. If a one-stage CRM is desired, second_stage_start_after should equal n in titecrm_args. |
| first_stage_label | A numeric value to be appended to all patients who belonged to the first stage. |
| sim_specific_start_id | A positive integer vector containing the starting subject id for each simulated trial. If provided, it must be as long as n_sim. |
| sim_specific_prior | If provided, this is a positive numeric matrix with number of rows equal to n_sim and number of columns equal to the number of dose levels, i.e. length(titecrm_args$PI). Each row gives the trial-specific skeleton to use. If not provided, then a common skeleton is used, taken from the value of titecrm_args$prior. |
| sim_specific_x0 | If provided, this is a non-negative integer vector with length equal to n_sim giving the starting dose level for each trial. If not provided, then a common starting dose is used, taken from the value of titecrm_args$x0. |
| sim_specific_scale | If provided, this is a positive numeric vector with length equal to n_sim giving the trial-specific value of the prior scale for beta in the power model p = skeleton ^ exp(beta). If not provided, then a common scale is used across simulations, taken from titecrm_args$scale. |
| seed | A positive integer seed for use prior to starting the simulations. |

## Value

The function returns the a named list containing:

**all_results** A matrix giving the individual patient outcomes from all simulated trials.

**first_stage_estMTD** A vector as long as the number of simulated trials giving an integer value corresponding to the estimated MTD from each trial as of the end of the first stage. A value of 0 indicates that all dose levels were estimated to be unsafe.

**second_stage_estMTD** The same result as first_stage_estMTD but at the end of the second stage.

**first_stage_enrollment** A vector as long as the number of simulated trials, giving the actual enrollment for that trial up to the end of the first stage.

**second_stage_enrollment** A vector as long as the number of simulated trials, giving the actual enrollment for that trial up to the end of the second stage. So the second stage result will include the enrollment from the first stage (and equal it if that simulated trial stopped in the first stage).

**seed** The seed that was used by the function.

`titecrm_phil` *Function to fit a TITE-CRM*

## Description

This is a function that makes dose recommendations for the next patient given the inputted data and the design parameters according to the continual reasssessment method. At the end of the trial, it returns estimates of the dose-toxicity curve under a one-parameter model where dose is the only predictor and prints out a dose recommendation. IMPORTANT: NO SAFETY CONSTRAINTS ARE IMPLEMENTED IN THIS FUNCTION. IT ONLY PRINTS OUT THE MODEL-BASED DOSE ASSIGNMENT FOR THE NEXT PATIENT; IT IS UP TO THE USER TO DETERMINE WHETHER ALL SAFETY CONSTRAINTS WOULD BE SATISFIED BY ANY GIVEN DOSE ASSIGNMENT AND TO REDUCE THE ASSIGNMENT AS NECESSARY. This function is based upon the titecrm function by Ken Cheung.

## Usage

```
titecrm_phil(
  prior,
  target,
  tox,
  level,
  n = length(level),
  weights = NULL,
  followup = NULL,
  entry = NULL,
  exit = NULL,
  obswin = NULL,
  scheme = "polynomial",
  scheme_args = list(scheme_power = 1),
  conf.level = 0.9,
  dosename = NULL,
  include = 1:n,
  pid = 1:n,
  method = "bayes",
  model = "empiric",
  var.est = TRUE,
  scale = sqrt(1.34),
  intcpt = 3,
  model.detail = TRUE,
  patient.detail = TRUE,
  tite = TRUE
)
```

## Arguments

| | |
|---|---|
| `prior` | A numeric vector with values between 0 and 1; the anticipated probabilities of toxicity for each dose. More commonly called the skeleton. |
| `target` | A scalar between 0 and 1 giving the targeted rate of DLT. |
| `tox` | An integer vector of 0s and 1s the same length as the current number of patients enrolled, indicating whether or not that patient had a toxicity. |

| | |
|---|---|
| level | An integer vector of dose numbers indicating dose assignments for all currently enrolled patients. Same length as tox. |
| n | An integer greater than 0 indicating the number of patients already enrolled, equal to the lengths of tox and level. |
| weights | A numeric vector of weights between 0 and 1 that control the likelihood contribution for each patient, in the situation where different patients are observed for different lengths of time. Same length as tox. |
| followup | A positive numeric vector indicating the number of units of time that each patient has been followed; same length as tox. |
| entry | Positive numeric vectors of entry and exit times; alternative to calculating followup. Same length as tox. |
| exit | Positive numeric vectors of entry and exit times; alternative to calculating followup. Same length as tox. |
| obswin | A positive numeric value indicating the number of units of time over which DLTs are defined. |
| scheme | A string indicating the weighting scheme for patients who are free of DLT but have not completed followup. Must be either "polynomial", "logistic", or "adaptive". "polynomial" is the default. |
| scheme_args | A named list with elements "scheme_power" (if "scheme" = "polynomial"), "scheme_int" and "scheme_slope" (if "scheme" = "logistic"), or no elements (if "scheme" = "adaptive"). |
| conf.level | A number between 0 and 1; the confidence limits to report. Default is 0.9. |
| dosename | A vector the same length as prior giving a list of names/identifiers for the different doses. |
| include | From titecrm documentation: "A subset of patients included in the dose calculation". Default is to include all patients. |
| pid | A vector of length n giving each patient's identifier. Default is to assign each patient an identifier from 1 to n. |
| method | A string indicating the method for fitting the model. The original titecrm function allows "mle" or "bayes"; titecrm_phil only includes "bayes". |
| model | A string indicating the type of model. The original titesim function allows "empiric" (sometimes known as the power model) or "logistic"; titecrm_phil only includes "empiric". |
| var.est | A logical value indicating if the posterior variance of model parameters should be returned. Default is TRUE. |
| scale | A positive numeric value indicating the prior standard deviation on the parameter beta. Default is the square root of 1.34. |
| intcpt | A fixed numeric value of the intercept parameter when using the "logistic" model. Default is 3. |
| model.detail | From titecrm documentation: "If FALSE, the model content of an mtd object will not be displayed. Default is TRUE". |
| patient.detail | From titecrm documentation: "If FALSE, patient summary of an mtd object will not be displayed. Default is TRUE". |
| tite | From titecrm documentation: "If FALSE, the time components in patient summary of an mtd object will be omitted. Default is TRUE". |

## Value

A named list with entries prior, target, tox, level, dosename, weights, followup, entry, exit, obswin, scheme, scheme_args, model, method, model.detail, intcpt, conf.level, include, tite, dosescaled, and patient.detail as described above, along with:

**prior.var** The prior variance of beta, or the user-inputted value scale squared.

**post.var** The posterior variance of beta.

**subset** A vector of patient IDs indicating which patients were included in the dose calculation.

**estimate** Posterior estimate of beta.

**mtd** Estimated MTD.

**ptox** Probability of toxicity at each dose.

**ptoxL** Lower confidence interval bound on the probability of toxicity at each dose.

**ptoxU** Upper confidence interval bound on the probability of toxicity at each dose.

**dosescaled** Scaled dose levels.

## References

Ken Cheung, "dfcrm: Dose-Finding by the Continual Reassessment Method." Version 0.2-2.1, 2019. https://CRAN.R-project.org/package=dfcrm

---

`titesim_phil` *Expanded version of the dfcrm::titesim function to incorporate some useful design elements of the time-to-event continual reassessment method*

---

## Description

This is the simulator function for a TITE-CRM trial. The user provides as input both the design elements as well as the true, dose-toxicity curve that is generally unknown in the real world. The simulator runs a certain number of simulated trials, creating data according to the dose-toxicity curve and making assignments according to the titecrm model. Various operating charateristics are reported. This function is meant to be called in the context of twostage_simulator rather than by the user directly.

## Usage

```
titesim_phil(
  PI,
  prior,
  target,
  n,
  x0,
  nsim = 1,
  restrict = TRUE,
  obswin = 1,
  tgrp = obswin,
  rate = 1,
  accrual = "fixed",
  surv = "uniform",
```

```
  surv_rate = obswin/10,
  scheme = "polynomial",
  scheme_args = list(scheme_power = 1),
  count = TRUE,
  method = "bayes",
  model = "empiric",
  intcpt = 3,
  scale = sqrt(1.34),
  seed = 1009,
  conf.level = 0.5,
  no.exceed = Inf,
  cohort.size = 1,
  first.cohort.only = T,
  n.at.MTD = Inf,
  followup_b4_esc = obswin,
  earliest_stop = 6
)
```

## Arguments

| | |
|---|---|
| PI | Numeric vector with entries between 0 and 1 of true toxicity probabilities; assumed to match the order of the dose labels. |
| prior | Numeric vector with entries between 0 and 1 of anticipated toxicity probabilities, assumed to match the order of the dose labels. More commonly and accurately called the skeleton. |
| target | Scalar value between 0 and 1 giving the targeted rate of DLT. |
| n | A positive integer indicating the maximum number of patients to enroll. |
| x0 | A positive integer indicating the starting dose level. |
| nsim | A positive integer indicating the number of simulated trials to conduct. |
| restrict | A logical value. If TRUE, indicates that safety constraints should be enacted. There are four: (1) No skipping doses in escalation; (2) No escalation before followup of followup_b4_esc on at least one patient at current or larger dose; (3) No assignment to dose with estimated DLT rate beyond no.exceed + target; (4) Stopping trial altogether if at any point after patient earliest_stop the estimated DLT rate at all dose levels exceeds no.exceed + target. |
| obswin | A positive number indicating the number of units of time over which DLTs are defined. |
| tgrp | This argument is an artefact of the dfcrm package and should be left blank. |
| rate | A positive number indicating the number of patients expected per unit obswin. |
| accrual | A string, either "fixed" or "poisson", denoting the patient accrual process. |
| surv | A string, either "uniform" or "exponential", denoting the time-to-DLT distribution. |
| surv_rate | A positive number. If surv is "exponential", this is the rate for the time-to-DLT distribution. |
| scheme | A string indicating the weighting scheme for patients who are free of DLT but have not completed followup. Can be either "polynomial", "logistic", or "adaptive." |

`scheme_args` A named list with elements "scheme power" (if scheme = "polynomial"), "scheme_int" and "scheme_slope" (if scheme = "logistic"), or no elements (if scheme = "adaptive").

`count` A logical value; if TRUE, the progress of the simulations will be plotted.

`method` A string indicating the method for fitting model. The original titesim function allows "mle" or "bayes"; here, only "bayes" is allowed.

`model` A string indicating the type of model. The original titesim function allows "empiric" (sometimes known as the power model) or "logistic"; here, only "empiric" is allowed.

`intcpt` A numeric value giving the intercept parameter when using the "logistic" model.

`scale` A numeric value giving the prior standard deviation on the parameter beta.

`seed` A positive integer random seed.

`conf.level` A number between 0 and 1 indicating the confidence limits to report.

`no.exceed` A positive number indicating by how much the toxicity rate can exceed the DLT threshold before the dose is unacceptable. To be more clear, no dose will be assigned with estimated pr(DLT) > target + no.exceed, and the trial will stop altogether if this holds for dose level 1.

`cohort.size` A positive integer indicating the size of cohorts between successive queries of the CRM model.

`first.cohort.only` A logical value indicating if cohort.size applies only to the first cohort.

`n.at.MTD` A positive integer which is the number of patients that should be at the estimated MTD before the trial stops. If n.at.MTD patients have been assigned to a single dose, and this dose is the current MTD that would be assigned to the next patient, then the trial stops.

`followup_b4_esc` A positive number indicating how much follow-up time at least one patient has to have on a given dose before the model escalates to the next dose.

`earliest_stop` A positive integer indicating how many patients must be observed before the trial can stop.

## Value

The function returns a list, with named components last_sim and all_sim. Access the last_sim to see the individual details of the single last simulation, or access all_sim to see summaries across all simulations performed.

---

# twostage_results *This function processes the output from twostage_simulator to produce several basic plots and summary tables.*

---

## Description

For more information on function inputs and features, please see the vignette and Boonstra (2020) on Arxiv.

## Usage

```
twostage_results(
  csv = FALSE,
  stage2folder = NULL,
  patientdatfolder = NULL,
  dose_outcome_curves_list = NULL,
  files = NULL,
  filepath = TRUE,
  primary_objectives = NULL,
  design_labels = NULL,
  scen_per_page = 10,
  design_per_page = 3,
  prop_label_size = 4,
  min_prop_to_write = 0.25,
  legend_text_size = 6,
  text_size = 9
)
```

## Arguments

| | |
|---|---|
| csv | A logical value indicating whether the output from twostage_simulator has been saved as .csv file(s) (if .csv, then TRUE). |
| stage2folder | A character string giving the filepath to a folder containing nothing but sim_data_stage2 output from twostage_simulator, saved as .csv file(s). This should only be provided if csv = TRUE. |
| patientdatfolder | A character string giving the filepath to a folder containing nothing but patient_data output from twostage_simulator, saved as .csv file(s). This should only be provided if csv = TRUE. |
| dose_outcome_curves_list | A list of lists, with each list element containing three named elements and an optional fourth element: tox_curve, eff_curve, scenario, and, optionally, eff_curve_stage2. tox_curve is the true toxicity curve of the doses; eff_curve is the true efficacy curve of the doses; scenario is an identifier of which true data-generating scenario is being run (meant to be helpful to the user when calling this function multiple times for different data-generating scenarios). This should be the same list that was used in twostage_simulator to generate all the simulated trials. This should only be provided if csv = TRUE. |
| files | A character string giving the filepath to a folder containing nothing but the raw output from twostage_simulator, saved as .Rds file(s). Alternatively, this can a character vector containing the names of all of the raw output from twostage_simulator, stored in the workspace. This should only be provided if csv = FALSE. |
| filepath | A logical value indicating whether files contains a filepath (filepath = TRUE) or a character vector of file names stored in the workspace (filepath = FALSE). The default is TRUE. This should only be provided if csv = FALSE. |
| primary_objectives | A list containing three named elements: tox_target, tox_delta_no_exceed, and eff_target, such that tox_target is between 0 and 1, tox_delta_no_exceed is between 0 and (1 - tox_target), and eff_target is between 0 and 1. This should be the same list that was used in twostage_simulator to generate the simulated trials. |

`design_labels` A character vector giving labels for the different designs. If left NULL, the number of the design will be used as its label. This should be the same length and the same order as design_list inputted in twostage_simulator.

`scen_per_page` A numeric value indicating the number of data-generating scenarios that should be printed per page. This can be at most 10 (the default is 10).

`design_per_page`
A numeric value indicating the number of designs that should be printed per page. This can be at most 3 (the default is 3).

`prop_label_size`
The value of the 'size' aesthetic in calls to geom_text (the default is 4).

`min_prop_to_write`
The smallest proportion to annotate in the stacked bar charts (the default is 0.25).

`legend_text_size`
The value of the 'size' argument for the legend elements in ggplots (the default is 6).

`text_size` The value of the 'size' argument for the other text elements in ggplots (the default is 9).

## Value

The function returns a named list containing two items: "plots" and "tables". tables contains:

**generating_params_for_display** This is a table giving the true efficacy and toxicity of each dose for each scenario considered.

**acc_dose_rec_table** This is a table giving which proportion of trials for each design and scenario combination recommended an acceptable dose (a dose meeting the toxicity and efficacy standards, if not the best dose) at the end of the trial.

**mean_patients_table** This is a table giving the mean number of patients enrolled for each design and scenario combination across the different simulated trials.

plots contains:

**gen_params_plot_blue** This is a list of plots providing the same information as in generating_params_for_display, in a blue color palette. If the number of designs under consideration exceeds design_per_page or the number of scenarios under consideration exceeds scen_per_page, the list will contain multiple plots. To call the first plot, we would run: results$plots$gen_param_plot[[1]] and so on for further plots.

**gen_params_plot_redgreen** This is a list of plots providing the same information as in generating_params_for_display, in a red-green color palette. If the number of designs under consideration exceeds design_per_page or the number of scenarios under consideration exceeds scen_per_page, the list will contain multiple plots. To call the first plot, we would run: results$plots$gen_param_plot[[1]] and so on for further plots.

**acc_dose_rec_plot** This is a plot giving which proportion of trials for each design and scenario combination recommended an acceptable dose, unacceptable dose, or made no recommendation at all. If the number of designs under consideration exceeds design_per_page or the number of scenarios under consideration exceeds scen_per_page, the list will contain multiple plots. To call the first plot, we would run: results$plots$acc_dose_rec_plot[[1]] and so on for further plots.

**dose_over_time_plot** This is a filled barplot giving the distribution of dose assignments by time, where time is measured by patient number. To call the first plot, we would run: results$plots$dose_over_time_plot[[
and so on for further plots.

**n_patients_plot** This is a boxplot of the number of patients enrolled for each design and scenario combination across the different simulated trials.

**n_patients_RP2D_plot** This is a boxplot of the number of patients who received the final recommended dose for each design and scenario combination across the different simulated trials. Note that the final recommended dose may not be safe or effective–this is merely a measure of how much patient data the design will yield for the final dose it recommends.

**prop_patients_accep_plot** This is a boxplot of the number of patients who received an acceptable dose (a dose meeting the toxicity and efficacy standards, if not the best dose) for each design and scenario combination across the different simulated trials.

---

`twostage_simulator` *This function simulates a specified number of seamless trials for each design configuration provided.*

---

## Description

twostage_simulator is the primary workhorse of seamlesssim. It simulates complex seamless Phase I/II oncology trials as discussed in the article by Boonstra et al. (Arxiv, 2020). It allows clinical trialists to determine operating characteristics of trials that assess both toxicity and efficacy with a range of different design and analytic approaches. For more detailed information, see Boonstra et al. (Arxiv, 2020) and the vignette.

## Usage

```
twostage_simulator(
  array_id = 1,
  n_sim,
  primary_objectives,
  dose_outcome_curves,
  design_list,
  stan_args = NA,
  sim_labels = NULL,
  design_labels = NULL,
  do_efficient_simulation = T,
  verbose = F,
  random_seed = 1,
  stan_seed = 1
)
```

## Arguments

| | |
|---|---|
| `array_id` | A positive integer identifier that will be appended as a column, without modification, to all results. This is meant to be helpful to the user when calling this function multiple times, e.g. in parallel. |
| `n_sim` | A positive integer indicating how many simulated trials to conduct for each design configuration. |
| `primary_objectives` | A list containing three named elements: tox_target, tox_delta_no_exceed, and eff_target, such that tox_target is between 0 and 1, tox_delta_no_exceed is between 0 and (1 - tox_target), and eff_target is between 0 and 1. These choices |

delineate the primary objectives of all designs to be simulated. The true MTD is defined as the dose level with true probability of DLT closest to tox_target but not exceeding tox_target + tox_delta_no_exceed, and true acceptable dose level(s) are defined as any dose level that is less than or equal to the MTD with a true probability of response at least as large as eff_target. Each design will recommend the estimated MTD if its estimated efficacy probability is at least eff_target, and otherwise recommend no dose level.

`dose_outcome_curves`

A list containing three named elements and an optional fourth element: tox_curve, eff_curve, scenario, and, optionally, eff_curve_stage2. tox_curve is the true toxicity curve of the doses; eff_curve is the true efficacy curve of the doses; scenario is an identifier of which true data-generating scenario is being run (meant to be helpful to the user when calling this function multiple times for different data-generating scenarios).

`design_list` A list specifying all specific module choices. It will be a list of lists of lists. The highest level of the list corresponds to each overall design to to be evaluated; this should be as long as the number of designs that the user wants to compare. The next level of the list gives the list of module choices for each design. It must have a named component module1 and will optionally have named components module2...module4, taken from the bolded values of Figure 1 in the manuscript referenced above. If any of module2 to module4 are not provided, they are assumed to correspond to a choice of module2 = list(name = "none"). Finally, the lowest level of the list gives the list of choices for each particular module. Each list must have one entry named "name" to indicate the choice of module, and also a value for every argument that is specific to that module. See the vignette for examples.

`stan_args` A list containing eight named elements for the Bayesian isotonic regression. For users without familiarity with STAN, stan_args can be left as NA (the default), and the defaults will all be used. Alternatively, users can modify any/all of these arguments, leaving the others as defaults or NA:

**n_mc_warmup** A positive integer giving the number of desired warmup runs; the default is 1000

**n_mc_samps** A positive integer giving the number of additional samples to run after warmup is completed; the default is 2000

**mc_chains** A positive integer indicating the number of chains to run in parallel, which will multiply the final number of samples; the default is 4

**mc_thins** A positive integer indicating the number of iterations to thin by (increasing thinning will decrease the final number of samples); the default is 1

**mc_stepsize** A numeric value between 0 and 1 that is passed to control in the call to stan() as the stepsize argument; the default is 0.1

**mc_adapt_delta** A numeric value between 0 and 1 that is passed to control in the call to stan() as the adapt_delta argument; the default is 0.8

**mc_max_treedepth** A positive integer passed to control in the call to stan() as the max_treedepth argument; the default is 15

**ntries** A positive integer. The stan algorithm throws warnings about divergent transitions, which are indicative of an inability to fully explore the posterior space. Sometimes this number can be extremely large, which suggests that the fitted model needs to be reparametrized. However, in this case, divergent transitions seem to be sporadic. ntries indicates how many reruns of

the algorithm should be tried when > 0 divergent transitions are encountered. The run with the fewest such transitions is kept. The default is 2.

**stan_seed** A positive integer used to randomly select the initial values for Stan sampling. The default is 1.

For users without familiarity with STAN who still wish to use Bayesian isotonic regression, some or all of these arguments may be left as NA, and default specifications will be used.

`sim_labels` A vector of anything but must be as long as n_sim. It will be included in the final data.frame of results under a column name of sim_id. It is provided to allow the user to uniquely identify simulations and is useful when this function is used in parallel.

`design_labels` A vector of anything but must be as long as length(design_list). It will be included in the final data.frame of results under a column name of design. It is provided to allow the user to uniquely identify designs and is useful when this function is used in parallel.

`do_efficient_simulation`
If TRUE, the simulator will run in such a way that, to the maximum possible extent, simulated data will be reused between consecutive designs. So, for example, design 1 may be identical to design 2 up to module 3, in which case the data from modules 1 and 2 can be reused from design 1 to design 2. If FALSE, each design will be simulated independently of each other design, but the whole simulator will take longer to run.

`verbose` If TRUE, the simulator will print all Stan and other function output; if FALSE, it will not. The default is FALSE.

`random_seed` A positive integer seed set prior to starting the simulations.

## Value

The function returns a named list with entries:

**patient_data** A data.frame with number of rows equal to number of individual patients simulated across all simulations of all designs, i.e. if every single design were to enroll the maximum possible number of patients, say, n, the number of rows would be n * length(design_list) * n_sim.

**sim_data_stage1** A data.frame with number of rows equal to length(design_list) * n_sim, i.e. one per design per simulation. It gives trial-level summary information about the status of the trial at the end of module 2 of each design.

**sim_data_stage2** A data.frame with number of rows equal to length(design_list) * n_sim, i.e. one per design per simulation. It gives trial-level summary information about the status of the trial at the end of module 4 of each design.

**dose_outcome_curves** The user-inputted argument to this function having the same name.

**titecrm_args** The list of common arguments that were used for the crm simulator.

**design_list** The user-inputted argument to this function having the same name.

**design_description** A character matrix with number of rows equal to length(design_list) and number of columns equal to the total number of modules used in the trial, presumably 4. It is meant to give a concise, simple summary and comparison of each design, without going into the details of each design.

**shared_design_elements** An integer matrix with number of rows equal to length(design_list) and number of columns equal to the total number of modules used in the trial, presumably 4. It gives the simulators assessment of which design elements could be recycled (therefore saving time if do_efficient_simulation==TRUE).

**random_seed** The user-inputted argument to this function having the same name.

# Index