Package: ewoc (via r-universe)

October 31, 2026

Type Package

Title Escalation with Overdose Control

Version 0.3.1

Description An implementation of a variety of escalation with overdose control designs introduced by Babb, Rogatko and Zacks (1998) <doi:10.1002/(SICI)1097-0258(19980530)17:10%3C1103::AID-SIM793%3E3.0.CO;2-9>. It calculates the next dose as a clinical trial proceeds and performs simulations to obtain operating characteristics.

License GPL (>= 2)

URL https://github.com/dnzmarcio/ewoc

BugReports https://github.com/dnzmarcio/ewoc/issues

Encoding UTF-8

Imports Formula(>= 1.2-1), rjags(>= 4-6), coda(>= 0.18-1), ggplot2(>= 2.2.0), graphics(>= 3.3.1), stats(>= 3.3.1), foreach(>= 1.4.3), doParallel(>= 1.0.11), doRNG (>= 1.7.1)

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Marcio A. Diniz [aut, cre, cph]

Maintainer Marcio A. Diniz <marcio.diniz@mountsinai.org>

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

Date/Publication 2026-08-31 22:30:22 UTC

RemoteUrl https://github.com/cran/ewoc

RemoteRef HEAD

RemoteSha c223977e3e031790c2e0601bdf1c24473ef88f9a

Contents

accuracy_index 2
average_toxicity 3

dlt_curve_d1classical 4
dlt_curve_d1extended 4
dlt_curve_d1ph 5
dlt_rate 6
ewoc_d1classical 7
ewoc_d1extended 10
ewoc_d1ph 12
ewoc_simulation 14
inv_standard_dose 17
logit 17
mtd_bias 18
mtd_mse 19
mtd_rho_d1extended 20
opc 20
optimal_mtd 23
optimal_toxicity 24
pdlt_d1classical 25
pdlt_d1extended 26
pdlt_d1ph 27
response_d1classical 28
response_d1extended 28
response_d1ph 29
standard_dose 30
stop_rule 31
stop_rule_d1classical 32
stop_rule_d1extended 33
stop_rule_d1ph 34

Index **36**

`accuracy_index` *Accuracy Index*

Description

Calculate the Accuracy Index.

Usage

```
accuracy_index(
  mtd_estimate,
  dose_set,
  true_prob,
  theta,
  loss = c("squared", "absolute", "classification", "overdose"),
  alpha = NULL
)
```

Arguments

mtd_estimate	a numerical vector of the MTD estimates.
dose_set	a numerical vector of allowable doses in the trial.
true_prob	a numerical vector of the true probabilities associated with 'dose_set'.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
loss	a loss function between the true probabilities of toxicity 'true_prob' and the target DLT rate 'theta'.
alpha	a numerical value indicating the weight of overdose for the overdose loss function.

Value

Accuracy Index for given loss function of the MTD estimates.

References

Cheung, Y. K. (2011). Dose finding by the continual reassessment method. CRC Press.

---

average_toxicity *Average Toxicity Number*

Description

Calculate the Average Toxicity Number.

Usage

```
average_toxicity(dose, dose_set, true_prob, theta)
```

Arguments

dose	a numerical matrix of assigned doses for each step of the trial (column) and for each trial (row).
dose_set	a numerical vector of allowable doses in the trial.
true_prob	a numerical vector of the true probabilities associated with 'dose_set'.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.

Value

Average Toxicity Number.

References

Cheung, Y. K. (2011). Dose finding by the continual reassessment method. CRC Press.

dlt_curve_d1classical *Plot the DLT curve based on the EWOC classical model*

Description

Plot the DLT curve based on the EWOC classical model

Usage

```
dlt_curve_d1classical(mtd, rho, theta, min_dose, max_dose, dose_set = NULL)
```

Arguments

mtd	a numerical value indicating the true value of the parameter mtd.
rho	a numerical value indicating the true value of the parameter rho.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.
dose_set	a numerical vector of doses that probability of DLT will be calculated.

dlt_curve_d1extended *Plot the DLT curve based on the EWOC extended model*

Description

Plot the DLT curve based on the EWOC extended model

Usage

```
dlt_curve_d1extended(rho, theta, min_dose, max_dose, dose_set = NULL)
```

Arguments

rho	a numerical vector indicating the true value of the parameters rho_0 and rho_1.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.
dose_set	a numerical vector of doses that probability of DLT will be calculated.

`dlt_curve_d1ph` *Plot the DLT curve based on the EWOC proportional hazards model*

Description

Plot the DLT curve based on the EWOC proportional hazards model

Usage

```
dlt_curve_d1ph(
  mtd,
  rho,
  theta,
  min_dose,
  max_dose,
  shape,
  tau,
  distribution = "exponential",
  dose_set = NULL
)
```

Arguments

`mtd`	a numerical value indicating the true value of the parameter mtd.
`rho`	a numerical value indicating the true value of the parameter rho.
`theta`	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
`min_dose`	a numerical value defining the lower bound of the support of the MTD.
`max_dose`	a numerical value defining the upper bound of the support of the MTD.
`shape`	a numerical value indicating the true value of the parameter shape. It is only necessary if 'distribution' = "weibull".
`tau`	a numerical value defining the period of time for a possible toxicity be observed.
`distribution`	a character establishing the distribution for the time of events. It can be defined as 'exponential' or 'weibull'.
`dose_set`	a numerical vector of doses that probability of DLT will be calculated.

`dlt_rate` *Evaluation of the DLT rate*

Description

Calculate the DLT rate for each trial, the average DLT rate, the percent of trials which have $DLTrate > target_rate + margin$, the percent of trials which have $DLTrate < target_rate - margin$ and the percent of trials which have $target_rate - margin < DLTrate < target_rate + margin$.

Usage

```
dlt_rate(
  dlt_matrix,
  trial = FALSE,
  target_rate = NULL,
  margin = NULL,
  digits = 2
)
```

Arguments

`dlt_matrix`	a matrix of the number of DLT for each step of the trial (column) and for each trial (row).
`trial`	a logical value indicating if the DLT rate for each trial should be returned.
`target_rate`	a numerical value of the target rate of DLT.
`margin`	a numerical value of the acceptable distance from the `target_rate`.
`digits`	a numerical value indicating the number of digits.

Value

`trial` a numerical vector of the DLT rate for each trial.

`average` a numerical value of the average of DLT rate considering a batch of trials.

`upper` the percent of trials which the `DLT rate` > `target_rate` + `margin` if `margin` != `NULL` and `target_rate` != `NULL`.

`lower` the percent of trials which the `DLT rate` < `target_rate` - `margin` if `margin` != `NULL` and `target_rate` != `NULL`.

`interval` the percent of trials which the `target_rate` - `margin` < `DLT rate` < `target_rate` + `margin` if `margin` != `NULL` and `target_rate` != `NULL`.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
stop_rule_sim(step_zero)
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 1, sample_size = 2,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       stop_rule_sim = stop_rule_sim,
                       ncores = 2)
dlt_rate(sim$dlt_sim)

## End(Not run)

## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
stop_rule_sim(step_zero)
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       stop_rule_sim = stop_rule_sim,
                       ncores = 2)
dlt_rate(sim$dlt_sim)

## End(Not run)
```

`ewoc_d1classical` *Escalation With Overdose Control*

Description

Finding the next dose for a phase I clinical trial based on the Escalation with Overdose Control (EWOC) design considering the classical parametrization for binary responses and single agent.

Usage

```
ewoc_d1classical(
  formula,
  theta,
  alpha,
  mtd_prior,
  rho_prior,
  min_dose,
  max_dose,
  type = c("continuous", "discrete"),
  first_dose = NULL,
  last_dose = NULL,
  dose_set = NULL,
  max_increment = NULL,
  no_skip_dose = TRUE,
  rounding = c("down", "nearest"),
  n_adapt = 5000,
  burn_in = 1000,
  n_mcmc = 1000,
  n_thin = 1,
  n_chains = 1
)
```

Arguments

formula	an object of class Formula: a symbolic description of the model to be fitted with only one regressor term corresponding to the dose for the right side and a numeric vector a response containing number of DLT for the left side.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
alpha	a numerical value defining the probability that the dose selected by EWOC is higher than the MTD.
mtd_prior	a matrix 1 x 2 of hyperparameters for the Beta prior distribution associated with the parameter MTD.
rho_prior	a matrix 1 x 2 of hyperparameters for the Beta prior distribution associated with the parameter rho.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.
type	a character describing the type of the Maximum Tolerable Dose (MTD) variable.
first_dose	a numerical value for the first allowable dose in the trial. It is only necessary if type = 'continuous'.

last_dose	a numerical value for the last allowable dose in the trial. It is only necessary if type = 'continuous'.
dose_set	a numerical vector of allowable doses in the trial. It is only necessary if type = 'discrete'.
max_increment	a numerical value indicating the maximum increment from the current dose to the next dose. It is only applied if type = 'continuous'.
no_skip_dose	a logical value indicating if it is allowed to skip doses. It is only necessary if type = 'discrete'. The default is TRUE.
rounding	a character indicating how to round a continuous dose to the one of elements of the dose set. It is only necessary if type = 'discrete'.
n_adapt	the number of iterations for adaptation. See adapt for details.
burn_in	numerical value indicating the number of iterations before to start monitoring.
n_mcmc	numerical value indicating the number of iterations to monitor.
n_thin	numerical value corresponding to the thinning interval for monitors.
n_chains	numerical value indicating the number of parallel chains for the model.

Value

next_dose the next recommend dose.

mtd the posterior MTD distribution.

rho the posterior rho_0 distribution.

sample a list of the MCMC chains distribution.

trial a list of the trial conditions.

References

Babb, J., Rogatko, A. and Zacks, S., 1998. Cancer phase I clinical trials: efficient dose escalation with overdose control. Statistics in medicine, 17(10), pp.1103-1120.

Examples

```
DLT <- 0
dose <- 20
test <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                      theta = 0.33, alpha = 0.25,
                      min_dose = 20, max_dose = 100,
                      dose_set = seq(20, 100, 20),
                      rho_prior = matrix(1, ncol = 2, nrow = 1),
                      mtd_prior = matrix(1, ncol = 2, nrow = 1),
                      rounding = "nearest")
summary(test)
plot(test)
```

ewoc_d1extended *Escalation With Overdose Control*

Description

Finding the next dose for a phase I clinical trial based on the Escalation with Overdose Control (EWOC) design considering the extended parametrization for binary response and single agent.

Usage

```
ewoc_d1extended(
  formula,
  theta,
  alpha,
  rho_prior,
  min_dose,
  max_dose,
  type = c("continuous", "discrete"),
  first_dose = NULL,
  last_dose = NULL,
  dose_set = NULL,
  max_increment = NULL,
  no_skip_dose = TRUE,
  rounding = c("down", "nearest"),
  n_adapt = 5000,
  burn_in = 1000,
  n_mcmc = 1000,
  n_thin = 1,
  n_chains = 1
)
```

Arguments

formula	an object of class Formula: a symbolic description of the model to be fitted with only one regressor term corresponding to the dose for the right side and a numeric vector as a response containing number of DLT for the left side.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
alpha	a numerical value defining the probability that the dose selected by EWOC is higher than the MTD.
rho_prior	a matrix 3 x 2 of hyperparameters for the Beta prior distribution associated with each parameter rho. Each row corresponds to a parameter.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.
type	a character describing the type of the Maximum Tolerable Dose (MTD) variable.

first_dose	a numerical value for the first allowable dose in the trial.
last_dose	a numerical value for the last allowable dose in the trial.
dose_set	a numerical vector of allowable doses in the trial. It is only necessary if type = "discrete".
max_increment	a numerical value indicating the maximum increment from the current dose to the next dose. It is only applied if type = 'continuous'.
no_skip_dose	a logical value indicating if it is allowed to skip doses. It is only necessary if type = 'discrete'. The default is TRUE.
rounding	a character indicating how to round a continuous dose to the one of elements of the dose set. It is only necessary if type = "discrete".
n_adapt	the number of iterations for adaptation. See adapt for details.
burn_in	the number of iterations before to start monitoring.
n_mcmc	the number of iterations to monitor.
n_thin	thinning interval for monitors.
n_chains	the number of parallel chains for the model.

Value

next_dose the next recommend dose.

mtd a numerical vector for the posterior MTD distribution considering the next patient covariable.

rho a matrix for the posterior rho_0 and rho_1 distributions.

sample a list of the MCMC chains distribution.

trial a list of the trial conditions.

References

Tighiouart, M., Cook-Wiens, G., & Rogatko, A. (2018). A Bayesian adaptive design for cancer phase I trials using a flexible range of doses. Journal of biopharmaceutical statistics, 28(3), 562-574.

Examples

```
DLT <- 0
dose <- 20

test <- ewoc_d1extended(DLT ~ dose, type = 'discrete',
                        theta = 0.33, alpha = 0.25,
                        dose_set = seq(20, 100, 20),
                        min_dose = 20, max_dose = 100,
                        rho_prior = matrix(1, ncol = 2, nrow = 2),
                        rounding = "nearest")
summary(test)
plot(test)
```

ewoc_d1ph *Escalation With Overdose Control*

Description

Finding the next dose for a phase I clinical trial based on Escalation with Overdose Control (EWOC) design considering parametrization for time to event response and single agent.

Usage

```
ewoc_d1ph(
  formula,
  theta,
  alpha,
  tau,
  type = c("continuous", "discrete"),
  rho_prior,
  mtd_prior,
  shape_prior = NULL,
  min_dose,
  max_dose,
  first_dose = NULL,
  last_dose = NULL,
  dose_set = NULL,
  max_increment = NULL,
  no_skip_dose = TRUE,
  distribution = c("exponential", "weibull"),
  rounding = c("down", "nearest"),
  n_adapt = 5000,
  burn_in = 1000,
  n_mcmc = 1000,
  n_thin = 1,
  n_chains = 1
)
```

Arguments

formula	an object of class Formula: a symbolic description of the model to be fitted with only one regressor term corresponding to the dose for the right side and a matrix as a response containing time and status for the left side.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
alpha	a numerical value defining the probability that the dose selected by EWOC is higher than the MTD.
tau	a numerical value defining the period of time for a possible toxicity be observed.

type	a character describing the type of the Maximum Tolerable Dose (MTD) variable. It can be 'discrete' or 'continuous'.
rho_prior	a matrix 1x2 of hyperparameters for the Beta prior distribution associated with the parameter rho.
mtd_prior	a matrix 1x2 of hyperparameters for the Beta prior distribution associated with the parameter MTD.
shape_prior	a matrix 1x2 of hyperparameters for the Gamma prior distribution associated with the shape parameter r for the Weibull distribution. It is only necessary if distribution = 'weibull'.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.
first_dose	a numerical value for the first allowable dose in the trial. It is only necessary if type = 'continuous'.
last_dose	a numerical value for the last allowable dose in the trial. It is only necessary if type = 'continuous'.
dose_set	a numerical vector of allowable doses in the trial. It is only necessary if type = 'discrete'.
max_increment	a numerical value indicating the maximum increment from the current dose to the next dose. It is only applied if type = 'continuous'.
no_skip_dose	a logical value indicating if it is allowed to skip doses. It is only necessary if type = 'discrete'. The default is TRUE.
distribution	a character establishing the distribution for the time of events. It can be defined as 'exponential' or 'weibull'.
rounding	a character indicating how to round a continuous dose to the one of elements of the dose set. It can be 'nearest' or 'down'. It is only necessary if type = 'discrete'.
n_adapt	the number of iterations for adaptation. See adapt for details.
burn_in	the number of iterations before to start monitoring.
n_mcmc	the number of iterations to monitor.
n_thin	thinning interval for monitors.
n_chains	the number of parallel chains for the model.

Value

next_dose the next recommend dose.

mtd the posterior MTD distribution.

rho the posterior rho_0 distribution.

sample a list of the MCMC chains distribution.

trial a list of the trial conditions.

References

Tighiouart M, Liu Y, Rogatko A. Escalation with overdose control using time to toxicity for cancer phase I clinical trials. PloS one. 2014 Mar 24;9(3):e93070.

Examples

```
time <- 9
status <- 0
dose <- 20

test <- ewoc_d1ph(cbind(time, status) ~ dose, type = 'discrete',
                  theta = 0.33, alpha = 0.25, tau = 10,
                  min_dose = 20, max_dose = 100,
                  dose_set = seq(20, 100, 20),
                  rho_prior = matrix(1, ncol = 2, nrow = 1),
                  mtd_prior = matrix(1, ncol = 2, nrow = 1),
                  distribution = 'exponential',
                  rounding = 'nearest')
summary(test)
plot(test)
```

---

ewoc_simulation *EWOC simulation*

Description

Generic function for simulating EWOC trials.

Usage

```
ewoc_simulation(
  step_zero,
  n_sim,
  sample_size,
  response_sim,
  fixed_first_cohort = TRUE,
  n_cohort = 1,
  alpha_strategy = "conditional",
  alpha_rate = 0.05,
  stop_rule_sim = NULL,
  ncores = 1,
  seed = 1234,
  ...
)
```

Arguments

step_zero	an object from the classes either 'ewoc_d1classical' or 'ewoc_d1extended' or 'ewoc_d1ph' created using the first cohort data.
n_sim	a number indicating the number of phase I clinical trials to be simulated.
sample_size	a number indicating the number of patients enrolled for each clinical trial.

`response_sim` a function which is self-contained and will be used as a generator function of the response variables in the simulation. Its only input is 'dose' and output is the indicator of DLT for classical and extended EWOC and the time until DLT for proportional hazards EWOC.

`fixed_first_cohort` a logical value indicating if the first cohort should be randomly generated or be fixed as the input in 'step_zero'.

`n_cohort` a number indicating the number of patients enrolled at each cohort. It is only used for 'ewoc_d1classical' and 'ewoc_d1extended'.

`alpha_strategy` a character indicating the strategy to apply for the feasibility value. Default is "constant". Options are "increasing" and "conditional".

`alpha_rate` a numerical value indicating the rate of the feasibility strategy. Only necessary if alpha_strategy is either 'increasing' or 'conditional'.

`stop_rule_sim` a function having as an input an object containing all the information related to the trial as the returned object trial from either `ewoc_d1classical`, `ewoc_d1extended`, `ewoc_d1ph` and as output a logical value indicating the trial should be stopped.

`ncores` a numeric value indicating the number of cores to be used in the simulation performed in parallel. Use parallel::detectCores() to check the number of cores available.

`seed` is an integer value, containing the random number generator (RNG) state for random number generation.

`...` For an object `step_zero` with class 'ewoc_d1ph', the argument `rate_sim` which controls the rate of accrue of patients following a Poisson process. The default is 1.

Value

`alpha_sim` a matrix `n_sim` x `sample_size` containing the values of feasibility used for each step in the trial and each trial in the simulation.

`dlt_sim` a matrix `n_sim` x `sample_size` containing ones and zeros indicating the occurrence of DLT (1) and the absence of DLT (0) for each step in the trial and each trial in the simulation.

`dose_sim` a matrix `n_sim` x `sample_size` containing the doses assigned for each step in the trial and each trial in the simulation.

`mtd_sim` a numeric vector `n_sim` x 1 containing the recommended MTD for each trial in the simulation.

`rho_sim` a numeric vector `n_sim` x k containing the estimated rho parameter(s) for each trial in the simulation, where k = 1 for ewoc_d1classical, ewoc_d1ph, and k = 2 for ewoc_d1extended.

Examples

```
## Not run:
### Classical EWOC
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
```

```
                          theta = 0.33, alpha = 0.25,
                          min_dose = 20, max_dose = 100,
                          dose_set = seq(20, 100, 20),
                          rho_prior = matrix(1, ncol = 2, nrow = 1),
                          mtd_prior = matrix(1, ncol = 2, nrow = 1),
                          rounding = "nearest")
response_sim <- response_d1classical(rho = 0.05, mtd = 60, theta = 0.33,
                                min_dose = 20, max_dose = 100)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30, n_cohort = 1,
                       alpha_strategy = "conditional",
                       response_sim = response_sim,
                       ncores = 1)

### Extended EWOC
DLT <- 0
dose <- 20
step_zero <- ewoc_d1extended(DLT ~ dose, type = 'discrete',
                          theta = 0.33, alpha = 0.25,
                          min_dose = 20, max_dose = 100,
                          dose_set = seq(20, 100, 20),
                          rho_prior = matrix(1, ncol = 2, nrow = 2),
                          rounding = "nearest")
response_sim <- response_d1extended(rho = c(0.05, 0.5),
                                 min_dose = 20, max_dose = 100)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30, n_cohort = 1,
                       alpha_strategy = "conditional",
                       response_sim = response_sim,
                       ncores = 1)

### PH EWOC
time <- 0
status <- 0
dose <- 20

step_zero <- ewoc_d1ph(cbind(time, status) ~ dose, type = 'discrete',
                       theta = 0.33, alpha = 0.25, tau = 10,
                       min_dose = 20, max_dose = 100,
                       dose_set = seq(20, 100, 20),
                       rho_prior = matrix(1, ncol = 2, nrow = 1),
                       mtd_prior = matrix(1, ncol = 2, nrow = 1),
                       distribution = 'exponential',
                       rounding = 'nearest')
response_sim <- response_d1ph(rho = 0.05, mtd = 60, theta = 0.33,
                              min_dose = 20, max_dose = 100,
                              tau = 10, distribution = "exponential")
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30, n_cohort = 1,
                       alpha_strategy = "conditional",
                       response_sim = response_sim,
                       ncores = 1)
```

```
## End(Not run)
```

`inv_standard_dose` *Inverse standardization of the dose*

Description

Unstandardizing a dose between minimum and maximum doses.

Usage

```
inv_standard_dose(dose, min_dose, max_dose)
```

Arguments

`dose`	a numerical value defining the standardized dose to be unstandardized.
`min_dose`	a numerical value defining the lower bound of the support of the MTD.
`max_dose`	a numerical value defining the upper bound of the support of the MTD.

Value

`dose` a numerical value between `min_dose` and `max_dose`.

`logit` *Logit*

Description

Calculating the logit of a probability

Usage

```
logit(p)
```

Arguments

`p`	a numerical value defining the probability to be transformed into logit scale.

Value

`logit` a numerical value in logit scale.

mtd_bias *Bias of the MTD estimates*

Description

Calculate the bias.

Usage

```
mtd_bias(mtd_estimate, true_mtd)
```

Arguments

mtd_estimate	a numerical vector of the MTD estimates.
true_mtd	a numerical value of the true Maximum Tolerable Dose.

Value

Bias of the MTD estimates.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       ncores = 2)
mtd_bias(sim$mtd_sim, true_mtd = 20)

## End(Not run)
```

`mtd_mse` *Mean Square Error of the MTD estimates*

Description

Calculate the Mean Square Error (MSE).

Usage

```
mtd_mse(mtd_estimate, true_mtd)
```

Arguments

`mtd_estimate`	a numerical vector of the MTD estimates.
`true_mtd`	a numerical value of the true Maximum Tolerable Dose.

Value

MSE of the MTD estimates.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       ncores = 2)
mtd_mse(sim$mtd_sim, true_mtd = 20)

## End(Not run)
```

mtd_rho_d1extended *Convert mtd to rho_1 and vice-versa*

Description

Converting mtd to rho_1 given rho_0 for EWOC design using extended parametrization and vice-versa.

Usage

```
mtd_rho_d1extended(mtd = NULL, rho_1 = NULL, rho_0, theta, min_dose, max_dose)
```

Arguments

mtd	a numerical value defining the mtd value to be converted into rho_1 value.
rho_1	a numerical value defining the probability of DLT at the max_dose.
rho_0	a numerical value defining the probability of DLT at the min_dose.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.

Value

mtd a numerical value for the maximum tolerable dose.

rho_1 a numerical value for the probability of DLT at the max_dose.

opc *Operating characteristics for EWOC simulations*

Description

Generic operating characteristics for one or more scenarios in EWOC simulations.

Usage

```
opc(sim_list, pdlt_list, mtd_list, toxicity_margin = NULL, mtd_margin = NULL)
```

Arguments

sim_list	a list of 'ewoc_simulation' objects for different scenarios created using the ewoc_simulation function.
pdlt_list	a list of functions to calculate the probability of toxicity with a numeric vector of doses as input and a numeric vector of probabilities as output.
mtd_list	a list of numerical values indicating the true MTD for each scenario.
toxicity_margin	a numerical value of the acceptable margin of distance from the target_rate.
mtd_margin	a numerical value of the acceptable margin of distance from the mtd_list.

Value

dlt_rate See dlt_rate.

dose_toxicity See optimal_toxicity.

mtd_toxicity See optimal_toxicity.

bias_mse See mtd_bias and mtd_mse.

dose_efficiency See optimal_mtd.

mtd_efficiency See optimal_mtd.

stop See stop_rule.

References

Diniz, M. A., Tighiouart, M., & Rogatko, A. (2019). Comparison between continuous and discrete doses for model based designs in cancer dose finding. PloS one, 14(1).

Examples

```
## Not run:
### Only one simulation
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                          theta = 0.33, alpha = 0.25,
                          min_dose = 20, max_dose = 100,
                          dose_set = seq(20, 100, 20),
                          rho_prior = matrix(1, ncol = 2, nrow = 1),
                          mtd_prior = matrix(1, ncol = 2, nrow = 1),
                          rounding = "nearest")
response_sim <- response_d1classical(rho = 0.05, mtd = 60, theta = 0.33,
                                  min_dose = 20, max_dose = 100)
sim <- ewoc_simulation(step_zero = step_zero,
                     n_sim = 1, sample_size = 30, n_cohort = 1,
                     alpha_strategy = "conditional",
                     response_sim = response_sim,
                     fixed_first_cohort =  TRUE,
```

```
                     ncores = 1)

pdlt <- pdlt_d1classical(rho = 0.05, mtd = 60, theta = 0.33,
                      min_dose = 20, max_dose = 100)

opc(sim_list = list(sim), pdlt_list = list(pdlt),
   mtd_list = list(60), toxicity_margin = 0.05, mtd_margin = 6)

### Two or more simulations

sim_list <- list()
mtd_list <- list()
pdlt_list <- list()

DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                          theta = 0.33, alpha = 0.25,
                          min_dose = 20, max_dose = 100,
                          dose_set = seq(20, 100, 20),
                          rho_prior = matrix(1, ncol = 2, nrow = 1),
                          mtd_prior = matrix(1, ncol = 2, nrow = 1),
                          rounding = "nearest")
mtd_list[[1]] <- 60
response_sim <- response_d1classical(rho = 0.05, mtd = mtd_list[[1]],
                                theta = 0.33,
                                min_dose = 20, max_dose = 100)
sim_list[[1]] <- ewoc_simulation(step_zero = step_zero,
                     n_sim = 1, sample_size = 30, n_cohort = 1,
                     alpha_strategy = "conditional",
                     response_sim = response_sim,
                     fixed_first_cohort =  TRUE,
                     ncores = 1)
pdlt_list[[1]] <- pdlt_d1classical(rho = 0.05, mtd = mtd_list[[1]],
                               theta = 0.33,
                               min_dose = 20, max_dose = 100)

mtd_list[[2]] <- 40
response_sim <- response_d1classical(rho = 0.05, mtd = mtd_list[[2]],
                                theta = 0.33,
                                min_dose = 20, max_dose = 100)
sim_list[[2]] <- ewoc_simulation(step_zero = step_zero,
                     n_sim = 1, sample_size = 30, n_cohort = 1,
                     alpha_strategy = "conditional",
                     response_sim = response_sim,
                     fixed_first_cohort =  TRUE,
                     ncores = 1)

pdlt_list[[2]] <- pdlt_d1classical(rho = 0.05, mtd = mtd_list[[2]],
                               theta = 0.33,
                               min_dose = 20, max_dose = 100)

opc(sim_list = sim_list, pdlt_list = pdlt_list,
```

```
   mtd_list = mtd_list, toxicity_margin = 0.05, mtd_margin = 6)

## End(Not run)
```

---

optimal_mtd *Percent of doses in relation the optimal MTD interval*

Description

Calculate the percent of doses which are inside the optimal MTD interval [true_MTD -margin ; true_MTD + margin].

Usage

```
optimal_mtd(dose_matrix, true_mtd, margin, digits = 2)
```

Arguments

dose_matrix	a numerical matrix or vector of assigned doses for each step of the trial (column) and for each trial (row).
true_mtd	a numerical value of the true Maximum Tolerable Dose.
margin	a numerical value of the acceptable margin of distance from the true_mtd.
digits	a numerical value indicating the number of digits.

Value

interval the average percent of doses which are inside the optimal MTD interval.

underdose the average percent of doses which are smaller than the lower limit of the optimal MTD interval.

overdose the average percent of doses which are greater than the upper limit of the optimal MTD interval.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
```

```
                                  min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       ncores = 2)
optimal_mtd(sim$mtd_sim, true_mtd = 20, margin = 0.1*20)
optimal_mtd(sim$dose_sim, true_mtd = 20, margin = 0.1*20)

## End(Not run)
```

`optimal_toxicity` *Percent of doses in relation the optimal toxicity interval*

Description

Calculate the percent of doses which are inside the optimal toxicity interval [`target rate -margin ; target rate + margin`].

Usage

```
optimal_toxicity(dose_matrix, target_rate, margin, pdlt, digits = 2)
```

Arguments

`dose_matrix`	a numerical matrix of assigned doses for each step of the trial (column) and for each trial (row).
`target_rate`	a numerical value of the target DLT rate.
`margin`	a numerical value of the acceptable margin of distance from the `target_rate`.
`pdlt`	a function to calculate the probability of toxicity with a numeric vector of doses as input and a numeric vector of probabilities as output.
`digits`	a numerical value indicating the number of digits.

Value

`interval` the average percent of doses which are inside the optimal toxicity interval.

`underdose` the average percent of doses which are smaller than the lower limit of the optimal toxicity interval.

`overdose` the average percent of doses which are greater than the upper limit of the optimal toxicity interval.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
pdlt_sim <- pdlt_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                             min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       ncores = 2)
optimal_toxicity(sim$mtd_sim, target_rate = 0.33, margin = 0.05, pdlt = pdlt_sim)
optimal_toxicity(sim$dose_sim, target_rate = 0.33, margin = 0.05, pdlt = pdlt_sim)

## End(Not run)
```

`pdlt_d1classical` *Generating a probability of DLT function based on the EWOC classical model*

Description

Generating a probability of DLT function based on the EWOC classical model

Usage

```
pdlt_d1classical(rho, mtd, theta, min_dose, max_dose)
```

Arguments

`rho`	a numerical value indicating the true value of the parameter rho.
`mtd`	a numerical value indicating the true value of the parameter mtd.
`theta`	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
`min_dose`	a numerical value defining the lower bound of the support of the MTD.
`max_dose`	a numerical value defining the upper bound of the support of the MTD.

Value

A function with dose as an input and a probability based on the logistic regression and parameters as an output.

Examples

```
pdlt <- pdlt_d1classical(rho = 0.05, mtd = 60, theta = 0.33,
                      min_dose = 20, max_dose = 100)

pdlt(20)
```

---

pdlt_d1extended *Generating a probability of DLT function based on the EWOC extended model*

Description

Generating a probability of DLT function based on the EWOC extended model

Usage

```
pdlt_d1extended(rho, min_dose, max_dose)
```

Arguments

rho	a numerical vector indicating the true value of the parameters rho_0 and rho_1.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.

Value

A function with dose as an input and a probability based on the logistic regression and parameters as an output.

Examples

```
pdlt <- pdlt_d1extended(rho = c(0.05, 0.5),
                      min_dose = 10, max_dose = 50)
pdlt(20)
```

pdlt_d1ph *Generating a probability of DLT function based on the EWOC Proportional Hazards model*

Description

Generating a probability of DLT function based on the EWOC Proportional Hazards model

Usage

```
pdlt_d1ph(rho, mtd, shape = NULL, theta, min_dose, max_dose, tau, distribution)
```

Arguments

rho	a numerical value indicating the true value of the parameter rho.
mtd	a numerical value indicating the true value of the parameter mtd.
shape	a numerical value indicating the true value of the parameter shape. It is only necessary if 'distribution' = "weibull".
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.
tau	a numerical value defining the period of time for a possible toxicity be observed.
distribution	a character establishing the distribution for the time of events. It can be defined as 'exponential' or 'weibull'.

Value

A function with dose as an input and a probability based on the logistic regression and parameters as an output.

Examples

```
pdlt <- pdlt_d1ph(rho = 0.05, mtd = 40, theta = 0.33,
                  min_dose = 30, max_dose = 50,
                  tau = 10, distribution = "exponential")
pdlt(40)
```

response_d1classical *Generating a binary response function based on the EWOC classical model*

Description

Generating a binary response function based on the EWOC classical model

Usage

```
response_d1classical(rho, mtd, theta, min_dose, max_dose)
```

Arguments

rho	a numerical value indicating the true value of the parameter rho.
mtd	a numerical value indicating the true value of the parameter mtd.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.

Value

A function with dose as an input and a Binomial variable based on the parameters as an output.

Examples

```
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
response_sim(20)
```

response_d1extended *Generating a binary response function based on the EWOC extended model*

Description

Generating a binary response function based on the EWOC extended model

Usage

```
response_d1extended(rho, min_dose, max_dose)
```

Arguments

rho	a numerical vector indicating the true value of the parameters rho_0 and rho_1.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.

Value

A function with dose as an input and a Binomial variable based on the parameters as an output.

Examples

```
response_sim <- response_d1extended(rho = c(0.05, 0.5),
                                    min_dose = 10, max_dose = 50)
response_sim(20)
```

---

response_d1ph *Generating a response function based on the EWOC Proportional Hazards model*

Description

Generating a response function based on the EWOC Proportional Hazards model

Usage

```
response_d1ph(
  rho,
  mtd,
  theta,
  min_dose,
  max_dose,
  tau,
  distribution,
  shape = NULL
)
```

Arguments

rho	a numerical value indicating the true value of the parameter rho.
mtd	a numerical value indicating the true value of the parameter mtd.
theta	a numerical value defining the proportion of expected patients to experience a medically unacceptable, dose-limiting toxicity (DLT) if administered the MTD.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.

tau	a numerical value defining the period of time for a possible toxicity be observed.
distribution	a character establishing the distribution for the time of events.
shape	a numerical value indicating the true value of the parameter shape. It is only necessary if 'distribution' = "weibull".

Value

A function with dose as an input and a Binomial variable based on the parameters as an output.

Examples

```
response_sim <- response_d1ph(rho = 0.05, mtd = 40, theta = 0.33,
                              min_dose = 30, max_dose = 50,
                              tau = 10, distribution = "exponential")
response_sim(40)
```

---

standard_dose *Standardization of the dose*

Description

Standardizing a dose between 0 and 1.

Usage

```
standard_dose(dose, min_dose, max_dose)
```

Arguments

dose	a numerical value defining the dose to be standardized.
min_dose	a numerical value defining the lower bound of the support of the MTD.
max_dose	a numerical value defining the upper bound of the support of the MTD.

Value

standardized dose a numerical value between 0 and 1.

stop_rule *Evaluation of the stopping rule*

Description

Calculate the average, minimum, maximum number of patients to stop a trial and the percent of stopped trials. Stopped trials contain NA after the last assigned dose.

Usage

```
stop_rule(dlt_matrix, sample_size, digits = 2)
```

Arguments

dlt_matrix	Matrix of the number of DLT for each step of the trial (column) and for each trial (row).
sample_size	a numerical value indicating the expected sample size.
digits	a numerical value indicating the number of digits.

Value

A list consisting of

average: Average number of patients to stop a trial.

min: Minimum number of patients to stop a trial.

max: Maximum number of patients to stop a trial.

nstop: Percent of stopped trials.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                          theta = 0.33, alpha = 0.25,
                          min_dose = 0, max_dose = 100,
                          dose_set = seq(0, 100, 20),
                          rho_prior = matrix(1, ncol = 2, nrow = 1),
                          mtd_prior = matrix(1, ncol = 2, nrow = 1),
                          rounding = "nearest")
stop_rule_sim(step_zero)
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                 min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 1, sample_size = 2,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       stop_rule_sim = stop_rule_sim,
```

```
                        ncores = 2)
stop_rule(sim$dlt_sim)

## End(Not run)

## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
stop_rule_sim(step_zero)
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "increasing",
                       response_sim = response_sim,
                       stop_rule_sim = stop_rule_sim,
                       ncores = 2)
stop_rule(sim$dlt_sim)

## End(Not run)
```

`stop_rule_d1classical` *Generating a stopping rule function for EWOC classical model*

Description

Generating a stopping rule function for EWOC classical model

Usage

```
stop_rule_d1classical(step)
```

Arguments

`step`	an object from the class 'ewoc_d1classical'.

Details

The function is evaluated at each step of the trial. It can defined based on any information contained in the object `step` that is the output from one of the functions `ewoc_d1classical`.

Value

a logical character indicating if the trial should be stopped or not.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1classical(DLT ~ dose, type = 'discrete',
                              theta = 0.33, alpha = 0.25,
                              min_dose = 0, max_dose = 100,
                              dose_set = seq(0, 100, 20),
                              rho_prior = matrix(1, ncol = 2, nrow = 1),
                              mtd_prior = matrix(1, ncol = 2, nrow = 1),
                              rounding = "nearest")
stop_rule_d1classical(step_zero)
response_sim <- response_d1classical(rho = 0.05, mtd = 20, theta = 0.33,
                                     min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "conditional",
                       response_sim = response_sim,
                       stop_rule_sim = stop_rule_d1_classical,
                       ncores = 1)

## End(Not run)
```

---

stop_rule_d1extended *Generating a stopping rule function for EWOC extended model*

Description

Generating a stopping rule function for EWOC extended model

Usage

```
stop_rule_d1extended(step)
```

Arguments

step	an object from the class ewoc_d1extended.

Details

The function is evaluated at each step of the trial. It can defined based on any information contained in the object step that is the output from one of the functions ewoc_d1extended.

Value

a logical character indicating if the trial should be stopped or not.

Examples

```
## Not run:
DLT <- 0
dose <- 20
step_zero <- ewoc_d1extended(DLT ~ dose, type = 'discrete',
                             theta = 0.33, alpha = 0.25,
                             min_dose = 0, max_dose = 100,
                             dose_set = seq(0, 100, 20),
                             rho_prior = matrix(1, ncol = 2, nrow = 2),
                             rounding = "nearest")
stop_rule_d1extended(step_zero)
response_sim <- response_d1extended(rho = c(0.05, 0.95),
                                    min_dose = 10, max_dose = 50)
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "conditional",
                       response_sim = response_sim,
                       stop_rule_sim = stop_rule_d1extended,
                       ncores = 1)

## End(Not run)
```

---

stop_rule_d1ph *Generating a stopping rule function for EWOC proportional hazards model*

Description

Generating a stopping rule function for EWOC proportional hazards model

Usage

```
stop_rule_d1ph(step)
```

Arguments

step	an object from the class ewoc_d1ph.

Details

The function is evaluated at each step of the trial. It can defined based on any information contained in the object step that is the output from one of the functions ewoc_d1ph.

Value

a logical character indicating if the trial should be stopped or not.

Examples

```
## Not run:
time <- 9
status <- 0
dose <- 20
step_zero <- ewoc_d1ph(cbind(time, status) ~ dose, type = 'discrete',
                       theta = 0.33, alpha = 0.25, tau = 10,
                       min_dose = 20, max_dose = 100,
                       dose_set = seq(20, 100, 20),
                       rho_prior = matrix(1, ncol = 2, nrow = 1),
                       mtd_prior = matrix(1, ncol = 2, nrow = 1),
                       distribution = 'exponential',
                       rounding = 'nearest')
stop_rule_d1ph(step_zero)
response_sim <- response_d1ph(rho = 0.05, mtd = 20, theta = 0.33,
                              min_dose = 10, max_dose = 50,
                              tau = 10, distribution = "exponential")
sim <- ewoc_simulation(step_zero = step_zero,
                       n_sim = 2, sample_size = 30,
                       alpha_strategy = "conditional",
                       response_sim = response_sim,
                       stop_rule_sim = stop_rule_d1ph,
                       ncores = 1)

## End(Not run)
```

Index

accuracy_index, 2
adapt, *9*, *11*, *13*
average_toxicity, 3

dlt_curve_d1classical, 4
dlt_curve_d1extended, 4
dlt_curve_d1ph, 5
dlt_rate, 6, *21*

ewoc_d1classical, 7
ewoc_d1extended, 10
ewoc_d1ph, 12
ewoc_simulation, 14, *21*

Formula, *8*, *10*, *12*

inv_standard_dose, 17

logit, 17

mtd_bias, 18, *21*
mtd_mse, 19, *21*
mtd_rho_d1extended, 20

opc, 20
optimal_mtd, *21*, 23
optimal_toxicity, *21*, 24

pdlt_d1classical, 25
pdlt_d1extended, 26
pdlt_d1ph, 27

response_d1classical, 28
response_d1extended, 28
response_d1ph, 29

standard_dose, 30
stop_rule, *21*, 31
stop_rule_d1classical, 32
stop_rule_d1extended, 33
stop_rule_d1ph, 34