

# Package ‘NCC’

May 7, 2026

**Title** Simulation and Analysis of Platform Trials with Non-Concurrent Controls

**Version** 1.0

**Author** Pavla Krotka [aut, cre] (ORCID: <https://orcid.org/0000-0001-5727-4270>),  
Marta Bofill Roig [aut, ths] (ORCID: <https://orcid.org/0000-0002-4400-7541>),  
Katharina Hees [aut],  
Peter Jacko [aut],  
Dominic Magirr [aut],  
Martin Posch [ctb] (ORCID: <https://orcid.org/0000-0001-8499-8573>)

**Maintainer** Pavla Krotka <pavla.krotka@meduniwien.ac.at>

**Description** Design and analysis of flexible platform trials with non-concurrent controls. Functions for data generation, analysis, visualization and running simulation studies are provided. The implemented analysis methods are described in:  
Bofill Roig et al. (2022) <[doi:10.1186/s12874-022-01683-w](https://doi.org/10.1186/s12874-022-01683-w)>,  
Saville et al. (2022) <[doi:10.1177/17407745221112013](https://doi.org/10.1177/17407745221112013)> and  
Schmidli et al. (2014) <[doi:10.1111/biom.12242](https://doi.org/10.1111/biom.12242)>.

**URL** <https://pavlakrotka.github.io/NCC/>,  
<https://github.com/pavlakrotka/NCC>

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** rlang, stats, RBeST, rjags, ggplot2, lmerTest, parallel,  
doParallel, parallelly, foreach, iterators, spaMM, mgcv,  
splines, magick

**RoxygenNote** 7.1.2

**Suggests** rmarkdown, knitr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-03-03 09:10:10 UTC

## Contents

datasim_bin . . . . .	2
datasim_cont . . . . .	6
fixmodel_bin . . . . .	10
fixmodel_cal_bin . . . . .	11
fixmodel_cal_cont . . . . .	13
fixmodel_cont . . . . .	15
gam_cont . . . . .	16
get_ss_matrix . . . . .	18
inv_u_trend . . . . .	19
linear_trend . . . . .	19
MAPprior_bin . . . . .	20
MAPprior_cont . . . . .	23
mixmodel_AR1_cal_cont . . . . .	26
mixmodel_AR1_cont . . . . .	27
mixmodel_cal_cont . . . . .	28
mixmodel_cont . . . . .	30
piecewise_cal_cont . . . . .	31
piecewise_cont . . . . .	33
plot_trial . . . . .	34
poolmodel_bin . . . . .	35
poolmodel_cont . . . . .	36
seasonal_trend . . . . .	37
sepmodel_adj_bin . . . . .	38
sepmodel_adj_cont . . . . .	39
sepmodel_bin . . . . .	41
sepmodel_cont . . . . .	42
sim_study . . . . .	43
sim_study_par . . . . .	45
splines_cal_cont . . . . .	47
splines_cont . . . . .	48
sw_trend . . . . .	50
timemachine_bin . . . . .	50
timemachine_cont . . . . .	53

<b>Index</b>	<b>56</b>
--------------	-----------

---

datasim_bin	<i>Simulate binary data from a platform trial with a shared control arm and a given number of experimental treatment arms entering at given time points</i>
-------------	---

---

### Description

This function simulates data from a platform trial with a given number of experimental treatment arms entering at given time points and a shared control arm. The primary endpoint is a binary endpoint. The user specifies the timing of adding arms in terms of patients recruited to the trial so far and the sample size per experimental treatment arm.

**Usage**

```

datasim_bin(
  num_arms,
  n_arm,
  d,
  period_blocks = 2,
  p0,
  OR,
  lambda,
  trend,
  N_peak,
  n_wave,
  full = FALSE,
  check = TRUE
)

```

**Arguments**

num_arms	Integer. Number of experimental treatment arms in the trial.
n_arm	Integer. Sample size per experimental treatment arm (assumed equal).
d	Integer vector with timings of adding new arms in terms of number of patients recruited to the trial so far. The first entry must be 0, so that the trial starts with at least one experimental treatment arm, and the entries must be non-decreasing. The vector length equals num_arms.
period_blocks	Integer. Number to define the size of the blocks for the block randomization. The block size in each period equals period_blockstimes the number of active arms in the period (see Details). Default=2.
p0	Double. Response probability in the control arm.
OR	Double vector with treatment effects in terms of odds ratios for each experimental treatment arm compared to control. The elements of the vector (odds ratios) are ordered by the entry time of the experimental treatment arms (e.g., the first entry in the vector corresponds to the odds ratio of the first experimental treatment arm). The vector length equals num_arms.
lambda	Double vector with strength of time trend in each arm ordered by the entry time of the arms (e.g., the first entry in the vector corresponds to the time trend in the control arm, second entry to the time trend in the first experimental treatment arm). The vector length equals num_arms+1, as time trend in the control is also allowed.
trend	String indicating the time trend pattern ("linear", "linear_2", "stepwise", "stepwise_2", "inv_u" or "seasonal"). See Details for more information.
N_peak	Integer. Timepoint at which the inverted-u time trend switches direction in terms of overall sample size (i.e. after how many recruited participants the trend direction switches).
n_wave	Integer. Number of cycles (waves) should the seasonal trend have.

full	Logical. Indicates whether the output should be in form of a data frame with variables needed for the analysis only (FALSE) or in form of a list containing more information (TRUE). Default=FALSE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.

## Details

### Design assumptions:

- The simulated platform trial consists of a given number of experimental treatment arms (specified by the argument num\_arms) and one control arm that is shared across the whole platform.
- Participants are indexed by entry order, assuming that at each time unit exactly one participant is recruited and the time of recruitment and observation of the response are equal.
- All participants are assumed to be eligible for all arms in the trial, i.e. the same inclusion and exclusion criteria apply to all experimental and control arms.
- Equal sample sizes (given by parameter n\_arm) in all experimental treatment arms are assumed.
- The duration of the trial is divided into so-called periods, defined as time intervals bounded by distinct time points of any treatment arm entering or leaving the platform. Hence, multiple treatment arms entering or leaving at the same time point imply the start of only one additional period.
- Allocation ratio of 1:1:...:1 in each period. Furthermore, block randomization is used to assign patients to the active arms. Block size in each period = period\_blocks\* (number of active arms in the period).
- If the period sample size is not a multiple of the block size, arms for the remaining participants are chosen by sampling without replacement from a vector containing the indices of active arms replicated times ceiling(remaining sample size/number of active arms).

### Data generation:

The binary response  $y_j$  for patient  $j$  is generated according to:

$$g(E(y_j)) = \eta_0 + \sum_{k=1}^K \cdot I(k_j = k) + f(j)$$

where  $g(\cdot)$  is the logit link function, and  $\eta_0$  (logit function of parameter  $p_0$ ) and  $\theta_k$  (log of the parameter OR) are the log odds in the control arm and the log odds ratio of treatment  $k$ .  $K$  is the total number of treatment arms in the trial (parameter num\_arms) and  $k_j$  is an indicator of the treatment arm patient  $j$  is allocated to.

The function  $f(j)$  denotes the time trend, whose strength is indicated by  $\lambda_{k_j}$  (parameter lambda) and which can have the following patterns (parameter trend):

- **"linear"** - trend starts at the beginning of the trial and the log odds increases or decreases linearly with a slope of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1}$ , where  $N$  is the total sample size in the trial

- **"linear\_2"** - trend starts after the first period (i.e. there is no time trend in the first period) and the log odds increases or decreases linearly with a slope of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1}$ , where  $N$  is the total sample size in the trial
- **"stepwise"** - the log odds is constant in each period and increases or decreases by  $\lambda$  each time any treatment arm enters or leaves the trial (i.e. in each period), according to the function  $f(j) = \lambda_{k_j} \cdot (c_j - 1)$ , where  $c_j$  is an index of the period patient  $j$  was enrolled in
- **"stepwise\_2"** - the log odds is constant in each period and increases or decreases by  $\lambda$  each time a new treatment arm is added to the trial, according to the function  $f(j) = \lambda_{k_j} \cdot (w_j - 1)$ , where  $w_j$  is an indicator of how many treatment arms have already entered the ongoing trial, when patient  $j$  was enrolled
- **"inv\_u"** - the log odds increases up to the point  $N_p$  (parameter `N_peak`) and decreases afterwards, linearly with a slope of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1} (I(j \leq N_p) - I(j > N_p))$ , where  $N_p$  indicates the point at which the trend turns from positive to negative in terms of the sample size (note that for negative  $\lambda$ , the log odds ratio decreases first and increases after)
- **"seasonal"** - the log odds increases and decreases periodically with a magnitude of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \sin(\psi \cdot 2\pi \cdot \frac{j-1}{N-1})$ , where  $\psi$  indicates how many cycles should the time trend have (parameter `n_wave`)

Trials with no time trend can be simulated too, by setting all elements of the vector `lambda` to zero and choosing an arbitrary pattern.

## Value

Data frame: simulated trial data (if `full=FALSE`, i.e. default) with the following columns:

- `j` - patient recruitment index
- `response` - binary response for patient `j`
- `treatment` - index of the treatment patient `j` was allocated to
- `period` - index of the period patient `j` was recruited in

or List (if `full=TRUE`) containing the following elements:

- `Data` - simulated trial data, including an additional column `p` with the probability used for simulating the response for patient `j`
- `n_total` - total sample size in the trial
- `n_arm` - sample size per arm (assumed equal)
- `num_arms` - number of experimental treatment arms in the trial
- `d` - timings of adding new arms
- `SS_matrix` - matrix with the sample sizes per arm and per period
- `period_blocks` - number to multiply the number of active arms with, in order to get the block size per period
- `p0` - response probability in the control arm
- `OR` - odds ratios for each experimental treatment arm
- `lambda` - strength of time trend in each arm
- `time_dep_effect` - time dependent treatment effects for each experimental treatment arm (for computing the bias)
- `trend` - time trend pattern

**Author(s)**

Pavla Krotka, Marta Bofill Roig

**Examples**

```
head(datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise"))
```

---

datasim_cont	<i>Simulate continuous data from a platform trial with a shared control arm and a given number of experimental treatment arms entering at given time points</i>
--------------	---

---

**Description**

This function simulates data from a platform trial with a given number of experimental treatment arms entering at given time points and a shared control arm. The primary endpoint is a continuous endpoint. The user specifies the timing of adding arms in terms of patients recruited to the trial so far and the sample size per arm.

**Usage**

```
datasim_cont(
  num_arms,
  n_arm,
  d,
  period_blocks = 2,
  mu0 = 0,
  theta,
  lambda,
  sigma,
  trend,
  N_peak,
  n_wave,
  full = FALSE,
  check = TRUE
)
```

**Arguments**

num_arms	Integer. Number of experimental treatment arms in the trial.
n_arm	Integer. Sample size per experimental treatment arm (assumed equal).
d	Integer vector with timings of adding new arms in terms of number of patients recruited to the trial so far. The first entry must be 0, so that the trial starts with at least one experimental treatment arm, and the entries must be non-decreasing. The vector length equals num_arms.

period_blocks	Integer. Number to define the size of the blocks for the block randomization. The block size in each period equals <code>period_blockstimes</code> the number of active arms in the period (see Details). Default=2.
mu0	Double. Response in the control arm. Default=0.
theta	Double vector with treatment effects in terms of difference of means for each experimental treatment arm compared to control. The elements of the vector (treatment effects) are ordered by the entry time of the experimental treatment arms (e.g., the first entry in the vector corresponds to the treatment effect of the first experimental treatment arm). The vector length equals <code>num_arms</code> .
lambda	Double vector with strength of time trend in each arm ordered by the entry time of the arms (e.g., the first entry in the vector corresponds to the time trend in the control arm, second entry to the time trend in the first experimental treatment arm). The vector length equals <code>num_arms+1</code> , as time trend in the control is also allowed.
sigma	Double. Standard deviation of the responses.
trend	String indicating the time trend pattern ("linear", "linear_2", "stepwise", "stepwise_2", "inv_u" or "seasonal"). See Details for more information.
N_peak	Integer. Timepoint at which the inverted-u time trend switches direction in terms of overall sample size (i.e. after how many recruited participants the trend direction switches).
n_wave	Integer. Number of cycles (waves) should the seasonal trend have.
full	Logical. Indicates whether the output should be in form of a data frame with variables needed for the analysis only (FALSE) or in form of a list containing more information (TRUE). Default=FALSE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.

## Details

### Design assumptions:

- The simulated platform trial consists of a given number of experimental treatment arms (specified by the argument `num_arms`) and one control arm that is shared across the whole platform.
- Participants are indexed by entry order, assuming that at each time unit exactly one participant is recruited and the time of recruitment and observation of the response are equal.
- All participants are assumed to be eligible for all arms in the trial, i.e. the same inclusion and exclusion criteria apply to all experimental and control arms.
- Equal sample sizes (given by parameter `n_arm`) in all experimental treatment arms are assumed.
- The duration of a platform trial is divided into so-called periods, defined as time intervals bounded by distinct time points of any treatment arm entering or leaving the platform. Hence, multiple treatment arms entering or leaving at the same time point imply the start of only one additional period.

- Allocation ratio of 1:1:...:1 in each period. Furthermore, block randomization is used to assign patients to the active arms. Block size in each period = period\_blocks\* (number of active arms in the period).
- If the period sample size is not a multiple of the block size, arms for the remaining participants are chosen by sampling without replacement from a vector containing the indices of active arms replicated times ceiling(remaining sample size/number of active arms).

### Data generation:

The continuous response  $y_j$  for patient  $j$  is generated according to:

$$E(y_j) = \eta_0 + \sum_{k=1}^K \cdot I(k_j = k) + f(j)$$

where  $\eta_0$  (parameter  $\mu\theta$ ) and  $\theta_k$  (parameter  $\theta_a$ ) are the response in the control arm and the effect of treatment  $k$ .  $K$  is the total number of treatment arms in the trial (parameter num\_arms) and  $k_j$  is an indicator of the treatment arm patient  $j$  is allocated to.

The function  $f(j)$  denotes the time trend, whose strength is indicated by  $\lambda_{k_j}$  (parameter lambda) and which can have the following patterns (parameter trend):

- **"linear"** - trend starts at the beginning of the trial and the mean response increases or decreases linearly with a slope of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1}$ , where  $N$  is the total sample size in the trial
- **"linear\_2"** - trend starts after the first period (i.e. there is no time trend in the first period) and the mean response increases or decreases linearly with a slope of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1}$ , where  $N$  is the total sample size in the trial
- **"stepwise"** - the mean response is constant in each period and increases or decreases by  $\lambda$  each time any treatment arm enters or leaves the trial (i.e. in each period), according to the function  $f(j) = \lambda_{k_j} \cdot (c_j - 1)$ , where  $c_j$  is an index of the period patient  $j$  was enrolled in
- **"stepwise\_2"** - the mean response is constant in each period and increases or decreases by  $\lambda$  each time a new treatment arm is added to the trial, according to the function  $f(j) = \lambda_{k_j} \cdot (w_j - 1)$ , where  $w_j$  is an indicator of how many treatment arms have already entered the ongoing trial, when patient  $j$  was enrolled
- **"inv\_u"** - the mean response increases up to the point  $N_p$  (parameter  $N_{peak}$ ) and decreases afterwards, linearly with a slope of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1} (I(j \leq N_p) - I(j > N_p))$ , where  $N_p$  indicates the point at which the trend turns from positive to negative in terms of the sample size (note that for negative  $\lambda$ , the mean response decreases first and increases after)
- **"seasonal"** - the mean response increases and decreases periodically with a magnitude of  $\lambda$ , according to the function  $f(j) = \lambda \cdot \sin(\psi \cdot 2\pi \cdot \frac{j-1}{N-1})$ , where  $\psi$  indicates how many cycles should the time trend have (parameter n\_wave)

Trials with no time trend can be simulated too, by setting all elements of the vector lambda to zero and choosing an arbitrary pattern.

**Value**

Data frame: simulated trial data (if full=FALSE, i.e. default) with the following columns:

- j - patient recruitment index
- response - continuous response for patient j
- treatment- index of the treatment patient j was allocated to
- period - index of the period patient j was recruited in

or List (if full=TRUE) containing the following elements:

- Data - simulated trial data, including an additional column means with the theoretical means used for the simulation of the response for patient j
- n\_total - total sample size in the trial
- n\_arm - sample size per arm (assumed equal)
- num\_arms - number of experimental treatment arms in the trial
- d - timings of adding new arms
- SS\_matrix - matrix with the sample sizes per arm and per period
- period\_blocks - number to multiply the number of active arms with, in order to get the block size per period
- mu0 - response in the control arm
- theta - treatment effects for each experimental treatment arm
- lambda - strength of time trend in each arm
- time\_dep\_effect - time dependent treatment effects for each experimental treatment arm (for computing the bias)
- sigma - standard deviation of the responses
- trend - time trend pattern

**Author(s)**

Pavla Krotka, Marta Bofill Roig

**Examples**

```
head(datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),  
theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear"))
```

---

fixmodel_bin	<i>Frequentist logistic regression model analysis for binary data adjusting for periods</i>
--------------	---

---

### Description

This function performs logistic regression taking into account all trial data until the arm under study leaves the trial and adjusting for periods as factors.

### Usage

```
fixmodel_bin(data, arm, alpha = 0.025, ncc = TRUE, check = TRUE, ...)
```

### Arguments

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

### Details

The model-based analysis adjusts for the time effect by including the factor period (defined as a time interval bounded by any treatment arm entering or leaving the platform). The time is then modelled as a step-function with jumps at the beginning of each period. Denoting by  $y_j$  the response probability for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$g(E(y_j)) = \eta_0 + \sum_{k \in \mathcal{K}_M} \theta_k \cdot I(k_j = k) + \sum_{s=2}^{S_M} \tau_s \cdot I(t_j \in T_{S_s})$$

where  $g(\cdot)$  denotes the logit link function and  $\eta_0$  is the log odds in the control arm in the first period;  $\theta_k$  represents the log odds ratio of treatment  $k$  and control for  $k \in \mathcal{K}_M$ , where  $\mathcal{K}_M$  is the set of treatments that were active in the trial during periods prior or up to the time when the investigated treatment arm left the trial;  $\tau_s$  indicates the stepwise period effect in terms of the log odds ratio between periods 1 and  $s$  ( $s = 2, \dots, S_M$ ), where  $S_M$  denotes the period, in which the investigated treatment arm left the trial.

If the data consists of only one period (e.g. in case of a multi-arm trial), the period is not used as covariate.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the log-odds ratio
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

**Author(s)**

Pavla Krotka

**References**

On model-based time trend adjustments in platform trials with non-concurrent controls. Bofill Roig, M., Krotka, P., et al. BMC Medical Research Methodology 22.1 (2022): 1-16.

**Examples**

```
trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")
```

```
fixmodel_bin(data = trial_data, arm = 3)
```

---

fixmodel_cal_bin	<i>Frequentist logistic regression model analysis for binary data adjusting for calendar time units</i>
------------------	---

---

**Description**

This function performs logistic regression taking into account all trial data until the arm under study leaves the trial and adjusting for calendar time units as factors.

**Usage**

```
fixmodel_cal_bin(  
  data,  
  arm,  
  alpha = 0.025,  
  unit_size = 25,  
  ncc = TRUE,  
  check = TRUE,  
  ...  
)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment' and 'response'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
unit_size	Integer. Number of patients per calendar time unit. Default=25.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The model-based analysis adjusts for the time effect by including the factor calendar time unit (defined as time units of fixed length, defined by `unit_size`). The time is then modelled as a step-function with jumps at the beginning of each calendar time unit. Denoting by  $y_j$  the response probability for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$g(E(y_j)) = \eta_0 + \sum_{k \in \mathcal{K}_M} \theta_k \cdot I(k_j = k) + \sum_{c=2}^{C_M} \tau_c \cdot I(t_j \in T_{C_c})$$

where  $g(\cdot)$  denotes the logit link function and  $\eta_0$  is the log odds in the control arm in the first calendar time unit;  $\theta_k$  represents the log odds ratio of treatment  $k$  and control for  $k \in \mathcal{K}_M$ , where  $\mathcal{K}_M$  is the set of treatments that were active in the trial during calendar time units prior or up to the time when the investigated treatment arm left the trial;  $\tau_c$  indicates the stepwise calendar time effect in terms of the log odds ratio between calendar time units 1 and  $c$  ( $c = 2, \dots, C_M$ ), where  $C_M$  denotes the calendar time unit, in which the investigated treatment arm left the trial.

If the data consists of only one calendar time unit, the calendar time unit is not used as covariate.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the log-odds ratio
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")
```

```
fixmodel_cal_bin(data = trial_data, arm = 3)
```

---

fixmodel_cal_cont	<i>Frequentist linear regression model analysis for continuous data adjusting for calendar time units</i>
-------------------	---

---

**Description**

This function performs linear regression taking into account all trial data until the arm under study leaves the trial and adjusting for calendar time units as factors.

**Usage**

```
fixmodel_cal_cont(
  data,
  arm,
  alpha = 0.025,
  unit_size = 25,
  ncc = TRUE,
  check = TRUE,
  ...
)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment' and 'response'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
unit_size	Integer. Number of patients per calendar time unit. Default=25.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

## Details

The model-based analysis adjusts for the time effect by including the factor calendar time unit (defined as time units of fixed length, defined by `unit_size`). The time is then modelled as a step-function with jumps at the beginning of each calendar time unit. Denoting by  $y_j$  the continuous response for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$E(y_j) = \eta_0 + \sum_{k \in \mathcal{K}_M} \theta_k \cdot I(k_j = k) + \sum_{c=2}^{C_M} \tau_c \cdot I(t_j \in T_{C_c})$$

where  $\eta_0$  is the response in the control arm in the first calendar time unit;  $\theta_k$  represents the effect of treatment  $k$  compared to control for  $k \in \mathcal{K}_M$ , where  $\mathcal{K}_M$  is the set of treatments that were active in the trial during calendar time units prior or up to the time when the investigated treatment arm left the trial;  $\tau_c$  indicates the stepwise calendar time effect between calendar time units 1 and  $c$  ( $c = 2, \dots, C_M$ ), where  $C_M$  denotes the calendar time unit, in which the investigated treatment arm left the trial.

If the data consists of only one calendar time unit, the calendar time unit is not used as covariate.

## Value

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)
- `model` - fitted model

## Author(s)

Pavla Krotka

## Examples

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

fixmodel_cal_cont(data = trial_data, arm = 3)
```

---

fixmodel_cont	<i>Frequentist linear regression model analysis for continuous data adjusting for periods</i>
---------------	---

---

### Description

This function performs linear regression taking into account all trial data until the arm under study leaves the trial and adjusting for periods as factors.

### Usage

```
fixmodel_cont(data, arm, alpha = 0.025, ncc = TRUE, check = TRUE, ...)
```

### Arguments

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

### Details

The model-based analysis adjusts for the time effect by including the factor period (defined as a time interval bounded by any treatment arm entering or leaving the platform). The time is then modelled as a step-function with jumps at the beginning of each period. Denoting by  $y_j$  the continuous response for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$E(y_j) = \eta_0 + \sum_{k \in \mathcal{K}_M} \theta_k \cdot I(k_j = k) + \sum_{s=2}^{S_M} \tau_s \cdot I(t_j \in T_{S_s})$$

where  $\eta_0$  is the response in the control arm in the first period;  $\theta_k$  represents the effect of treatment  $k$  compared to control for  $k \in \mathcal{K}_M$ , where  $\mathcal{K}_M$  is the set of treatments that were active in the trial during periods prior or up to the time when the investigated treatment arm left the trial;  $\tau_s$  indicates the stepwise period effect between periods 1 and  $s$  ( $s = 2, \dots, S_M$ ), where  $S_M$  denotes the period, in which the investigated treatment arm left the trial.

If the data consists of only one period (e.g. in case of a multi-arm trial), the period is not used as covariate.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the difference in means
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

**Author(s)**

Pavla Krotka

**References**

On model-based time trend adjustments in platform trials with non-concurrent controls. Bofill Roig, M., Krotka, P., et al. *BMC Medical Research Methodology* 22.1 (2022): 1-16.

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

fixmodel_cont(data = trial_data, arm = 3)
```

---

gam\_cont

*Generalized additive model analysis for continuous data*

---

**Description**

This function performs analysis using a generalized additive model taking into account all trial data until the arm under study leaves the trial and smoothing over the patient entry index.

**Usage**

```
gam_cont(
  data,
  arm,
  alpha = 0.025,
  ci = FALSE,
  smoothing_basis = "tp",
  basis_dim = -1,
  gam_method = "GCV.Cp",
  check = TRUE,
  ...
)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response', 'period' and 'j'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
ci	Logical. Indicates whether confidence intervals should be computed. Default=FALSE.
smoothing_basis	String indicating the (penalized) smoothing basis to use. Default="tp" for thin plate regression spline. Available strings are 'tp', 'ts', 'ds', 'cr', 'cs', 'cc', 'sos', 'ps', 'cp', 're', 'mrf', 'gp', and 'so'. For more information see <a href="https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/smooth.terms.html">https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/smooth.terms.html</a> .
basis_dim	Integer. The dimension of the basis used to represent the smooth term. The default depends on the number of variables that the smooth is a function of. Default=-1. For more information see the description of the parameter 'k' in <a href="https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/s.html">https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/s.html</a> .
gam_method	String indicating the smoothing parameter estimation method. Default="GCV.Cp". Available strings are 'GCV.Cp', 'GACV.Cp', 'REML', 'P-REML', 'ML', and 'P-ML'. For more information see the description of the parameter 'method' in <a href="https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/gam.html">https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/gam.html</a> .
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

gam_cont(data = trial_data, arm = 3, ci = TRUE)
```

---

get_ss_matrix	<i>Sample size matrix for a platform trial with a given number of treatment arms</i>
---------------	--

---

### Description

This function computes the matrix with sample sizes per arm and period. It is used in the functions `datasim_bin()` and `datasim_cont()`.

### Usage

```
get_ss_matrix(num_arms, n_arm, d)
```

### Arguments

<code>num_arms</code>	Integer. Number of experimental treatment arms in the trial.
<code>n_arm</code>	Integer. Sample size per experimental treatment arm.
<code>d</code>	Integer vector with timings of adding new arms in terms of number of patients recruited to the trial so far. The first entry must be 0, so that the trial starts with at least one experimental treatment arm, and the entries must be non-decreasing. The vector length equals <code>num_arms</code> .

### Value

Sample size matrix, consisting of the sample size per arm and per period, where the arms are represented in the rows (with the control arm in the first row and the experimental arms coming after ordered by entry time) and the periods are represented in the columns.

### Author(s)

Pavla Krotka

### Examples

```
get_ss_matrix(num_arms = 3, n_arm = 100, d = c(0, 100, 250))
```

---

inv\_u\_trend                      *Generation of an inverted-u trend*

---

**Description**

This function generates a time trend for given time points in the trial according to an inverted-u function.

**Usage**

```
inv_u_trend(j, lambda, N_peak, n_total)
```

**Arguments**

j	Time points for which the trend should be generated.
lambda	Strength of time trend.
N_peak	Point at which the time trend switches direction in terms of overall sample size.
n_total	Total sample size in the trial.

**Details**

The time trend is generated according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1} (I(j \leq N_p) - I(j > N_p))$ , where  $N$  is the total sample size (parameter n\_total) and  $N_p$  (parameter N\_peak) indicates the point at which the trend switches direction.

**Value**

Time trend for time points j.

**Author(s)**

Marta Bofill Roig, Pavla Krotka

---

linear\_trend                      *Generation of a linear trend that starts in a given period*

---

**Description**

This function generates a time trend for given time points in the trial according to a linear function.

**Usage**

```
linear_trend(j, lambda, sample_size)
```

**Arguments**

j	Time points for which the trend should be generated.
lambda	Strength of time trend.
sample_size	Vector of dimension 2, indicating sample size in the trial period until the time trend starts and the remaining sample size.

**Details**

The time trend is generated according to the function  $f(j) = \lambda \cdot \frac{j-1}{N-1}$ , where  $N$  is the total sample size.

**Value**

Time trend for time points j.

**Author(s)**

Marta Bofill Roig, Pavla Krotka

---

MAPprior\_bin

*Analysis for binary data using the MAP Prior approach*

---

**Description**

This function performs analysis of binary data using the Meta-Analytic-Predictive (MAP) Prior approach. The method borrows data from non-concurrent controls to obtain the prior distribution for the control response in the concurrent periods.

**Usage**

```
MAPprior_bin(  
  data,  
  arm,  
  alpha = 0.025,  
  opt = 2,  
  prior_prec_tau = 4,  
  prior_prec_eta = 0.001,  
  n_samples = 1000,  
  n_chains = 4,  
  n_iter = 4000,  
  n_adapt = 1000,  
  robustify = TRUE,  
  weight = 0.1,  
  check = TRUE,  
  ...  
)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Decision boundary (one-sided). Default=0.025
opt	Integer (1 or 2). If <code>opt==1</code> , all former periods are used as one source; if <code>opt==2</code> , periods get separately included into the final analysis. Default=2.
prior_prec_tau	Double. Precision parameter ( $1/\sigma_\tau^2$ ) of the half normal hyperprior, the prior for the between study heterogeneity. Default=4.
prior_prec_eta	Double. Precision parameter ( $1/\sigma_\eta^2$ ) of the normal hyperprior, the prior for the hyperparameter mean of the control log-odds. Default=0.001.
n_samples	Integer. Number of how many random samples will get drawn for the calculation of the posterior mean, the p-value and the CI's. Default=1000.
n_chains	Integer. Number of parallel chains for the rjags model. Default=4.
n_iter	Integer. Number of iterations to monitor of the jags.model. Needed for <code>coda.samples</code> . Default=4000.
n_adapt	Integer. Number of iterations for adaptation, an initial sampling phase during which the samplers adapt their behavior to maximize their efficiency. Needed for <code>jags.model</code> . Default=1000.
robustify	Logical. Indicates whether a robust prior is to be used. If TRUE, a mixture prior is considered combining a MAP prior and a weakly non-informative component prior. Default=TRUE.
weight	Double. Weight given to the non-informative component ( $0 < \text{weight} < 1$ ) for the robustification of the MAP prior according to Schmidli (2014). Default=0.1.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The MAP approach derives the prior distribution for the control response in the concurrent periods by combining the control information from the non-concurrent periods with a non-informative prior.

The model for the binary response  $y_{js}$  for the control patient  $j$  in the non-concurrent period  $s$  is defined as follows:

$$g(E(y_{js})) = \eta_s$$

where  $g(\cdot)$  denotes the logit link function and  $\eta_s$  represents the control log odds in the non-concurrent period  $s$ .

The log odds for the non-concurrent controls in period  $s$  are assumed to have a normal prior distribution with mean  $\mu_\eta$  and variance  $\tau^2$ :

$$\eta_s \sim \mathcal{N}(\mu_\eta, \tau^2)$$

For the hyperparameters  $\mu_\eta$  and  $\tau$ , normal and half-normal hyperprior distributions are assumed, with mean 0 and variances  $\sigma_\eta^2$  and  $\sigma_\tau^2$ , respectively:

$$\mu_\eta \sim \mathcal{N}(0, \sigma_\eta^2)$$

$$\tau \sim \text{HalfNormal}(0, \sigma_\tau^2)$$

The MAP prior distribution  $p_{MAP}(\eta_{CC})$  for the control response in the concurrent periods is then obtained as the posterior distribution of the parameters  $\eta_s$  from the above specified model.

If `robustify=TRUE`, the MAP prior is robustified by adding a weakly-informative mixture component  $p_{\text{non-inf}}$ , leading to a robustified MAP prior distribution:

$$p_{rMAP}(\eta_{CC}) = (1 - w) \cdot p_{MAP}(\eta_{CC}) + w \cdot p_{\text{non-inf}}(\eta_{CC})$$

where  $w$  (parameter weight) may be interpreted as the degree of skepticism towards borrowing strength.

In this function, the argument `alpha` corresponds to  $1 - \gamma$ , where  $\gamma$  is the decision boundary. Specifically, the posterior probability of the difference distribution under the null hypothesis is such that:  $P(p_{\text{treatment}} - p_{\text{control}} > 0) \geq 1 - \alpha$ . In case of a non-informative prior this coincides with the frequentist type I error.

## Value

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - posterior probability that the log-odds ratio is less than zero
- `treat_effect` - posterior mean of log-odds ratio
- `lower_ci` - lower limit of the  $(1 - 2 \cdot \alpha) \cdot 100\%$  credible interval for log-odds ratio
- `upper_ci` - upper limit of the  $(1 - 2 \cdot \alpha) \cdot 100\%$  credible interval for log-odds ratio
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)

## Author(s)

Katharina Hees

## References

Robust meta-analytic-predictive priors in clinical trials with historical control information. Schmidli, H., et al. *Biometrics* 70.4 (2014): 1023-1032.

Applying Meta-Analytic-Predictive Priors with the R Bayesian Evidence Synthesis Tools. Weber, S., et al. *Journal of Statistical Software* 100.19 (2021): 1548-7660.

**Examples**

```

trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")

MAPprior_bin(data = trial_data, arm = 3)

```

---

MAPprior\_cont

*Analysis for continuous data using the MAP Prior approach*


---

**Description**

This function performs analysis of continuous data using the Meta-Analytic-Predictive (MAP) Prior approach. The method borrows data from non-concurrent controls to obtain the prior distribution for the control response in the concurrent periods.

**Usage**

```

MAPprior_cont(
  data,
  arm,
  alpha = 0.025,
  opt = 2,
  prior_prec_tau = 4,
  prior_prec_eta = 0.001,
  n_samples = 1000,
  n_chains = 4,
  n_iter = 4000,
  n_adapt = 1000,
  robustify = TRUE,
  weight = 0.1,
  check = TRUE,
  ...
)

```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Decision boundary (one-sided). Default=0.025
opt	Integer (1 or 2). If <code>opt==1</code> , all former periods are used as one source; if <code>opt==2</code> , periods get separately included into the final analysis. Default=2.

prior_prec_tau	Double. Precision parameter ( $1/\sigma_\tau^2$ ) of the half normal hyperprior, the prior for the between study heterogeneity. Default=4.
prior_prec_eta	Double. Precision parameter ( $1/\sigma_\eta^2$ ) of the normal hyperprior, the prior for the hyperparameter mean of the control mean. Default=0.001.
n_samples	Integer. Number of how many random samples will get drawn for the calculation of the posterior mean, the p-value and the CI's. Default=1000.
n_chains	Integer. Number of parallel chains for the rjags model. Default=4.
n_iter	Integer. Number of iterations to monitor of the jags.model. Needed for coda.samples. Default=4000.
n_adapt	Integer. Number of iterations for adaptation, an initial sampling phase during which the samplers adapt their behavior to maximize their efficiency. Needed for jags.model. Default=1000.
robustify	Logical. Indicates whether a robust prior is to be used. If TRUE, a mixture prior is considered combining a MAP prior and a weakly non-informative component prior. Default=TRUE.
weight	Double. Weight given to the non-informative component ( $0 < \text{weight} < 1$ ) for the robustification of the MAP prior according to Schmidli (2014). Default=0.1.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

## Details

The MAP approach derives the prior distribution for the control response in the concurrent periods by combining the control information from the non-concurrent periods with a non-informative prior.

The model for the continuous response  $y_{js}$  for the control patient  $j$  in the non-concurrent period  $s$  is defined as follows:

$$E(y_{js}) = \eta_s$$

where  $\eta_s$  represents the control mean in the non-concurrent period  $s$ .

The means for the non-concurrent controls in period  $s$  are assumed to have a normal prior distribution with mean  $\mu_\eta$  and variance  $\tau^2$ :

$$\eta_s \sim \mathcal{N}(\mu_\eta, \tau^2)$$

For the hyperparameters  $\mu_\eta$  and  $\tau$ , normal and half-normal hyperprior distributions are assumed, with mean 0 and variances  $\sigma_\eta^2$  and  $\sigma_\tau^2$ , respectively:

$$\mu_\eta \sim \mathcal{N}(0, \sigma_\eta^2)$$

$$\tau \sim \text{HalfNormal}(0, \sigma_\tau^2)$$

The MAP prior distribution  $p_{MAP}(\eta_{CC})$  for the control response in the concurrent periods is then obtained as the posterior distribution of the parameters  $\eta_s$  from the above specified model.

If `robustify=TRUE`, the MAP prior is robustified by adding a weakly-informative mixture component  $p_{\text{non-inf}}$ , leading to a robustified MAP prior distribution:

$$p_{rMAP}(\eta_{CC}) = (1 - w) \cdot p_{MAP}(\eta_{CC}) + w \cdot p_{\text{non-inf}}(\eta_{CC})$$

where  $w$  (parameter weight) may be interpreted as the degree of skepticism towards borrowing strength.

In this function, the argument `alpha` corresponds to  $1 - \gamma$ , where  $\gamma$  is the decision boundary. Specifically, the posterior probability of the difference distribution under the null hypothesis is such that:  $P(\mu_{\text{treatment}} - \mu_{\text{control}} > 0) \geq 1 - \alpha$ . In case of a non-informative prior this coincides with the frequentist type I error.

### Value

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - posterior probability that the difference in means is less than zero
- `treat_effect` - posterior mean of difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  credible interval for difference in means
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  credible interval for difference in means
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)

### Author(s)

Katharina Hees

### References

Robust meta-analytic-predictive priors in clinical trials with historical control information. Schmidli, H., et al. *Biometrics* 70.4 (2014): 1023-1032.

Applying Meta-Analytic-Predictive Priors with the R Bayesian Evidence Synthesis Tools. Weber, S., et al. *Journal of Statistical Software* 100.19 (2021): 1548-7660.

### Examples

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "stepwise")
```

```
MAPprior_cont(data = trial_data, arm = 3)
```

---

`mixmodel_AR1_cal_cont` *Mixed regression model analysis for continuous data adjusting for calendar time units as a random factor with AR1 correlation structure*

---

### Description

This function performs linear mixed model regression taking into account all trial data until the arm under study leaves the trial and adjusting for calendar time units as random factors with AR1 correlation structure.

### Usage

```

mixmodel_AR1_cal_cont(
  data,
  arm,
  alpha = 0.025,
  ci = FALSE,
  unit_size = 25,
  ncc = TRUE,
  check = TRUE,
  ...
)

```

### Arguments

<code>data</code>	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment' and 'response'.
<code>arm</code>	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
<code>alpha</code>	Double. Significance level (one-sided). Default=0.025.
<code>ci</code>	Logical. Indicates whether confidence intervals should be computed. Default=FALSE.
<code>unit_size</code>	Integer. Number of patients per calendar time unit. Default=25.
<code>ncc</code>	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
<code>check</code>	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
<code>...</code>	Further arguments passed by wrapper functions when running simulations.

### Value

List containing the following elements regarding the results of comparing arm to control:

- `p-val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means

- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

### Author(s)

Pavla Krotka

### Examples

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

mixmodel_AR1_cal_cont(data = trial_data, arm = 3, ci = TRUE)
```

---

mixmodel_AR1_cont	<i>Mixed regression model analysis for continuous data adjusting for periods as a random factor with AR1 correlation structure</i>
-------------------	--

---

### Description

This function performs linear mixed model regression taking into account all trial data until the arm under study leaves the trial and adjusting for periods as random factors with AR1 correlation structure.

### Usage

```
mixmodel_AR1_cont(
  data,
  arm,
  alpha = 0.025,
  ci = FALSE,
  ncc = TRUE,
  check = TRUE,
  ...
)
```

### Arguments

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.

ci	Logical. Indicates whether confidence intervals should be computed. Default=FALSE.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

### Value

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the difference in means
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

### Author(s)

Pavla Krotka

### Examples

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

mixmodel_AR1_cont(data = trial_data, arm = 3, ci = TRUE)
```

---

mixmodel_cal_cont	<i>Mixed regression model analysis for continuous data adjusting for calendar time units as a random factor</i>
-------------------	---

---

### Description

This function performs linear mixed model regression taking into account all trial data until the arm under study leaves the trial and adjusting for calendar time units as random factors.

**Usage**

```

mixmodel_cal_cont(
  data,
  arm,
  alpha = 0.025,
  ci = FALSE,
  unit_size = 25,
  ncc = TRUE,
  check = TRUE,
  ...
)

```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment' and 'response'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
ci	Logical. Indicates whether confidence intervals should be computed. Default=FALSE.
unit_size	Integer. Number of patients per calendar time unit. Default=25.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```

trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

mixmodel_cal_cont(data = trial_data, arm = 3, ci = TRUE)

```

---

mixmodel_cont	<i>Mixed regression model analysis for continuous data adjusting for periods as a random factor</i>
---------------	---

---

**Description**

This function performs linear mixed model regression taking into account all trial data until the arm under study leaves the trial and adjusting for periods as random factors.

**Usage**

```

mixmodel_cont(
  data,
  arm,
  alpha = 0.025,
  ci = FALSE,
  ncc = TRUE,
  check = TRUE,
  ...
)

```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
ci	Logical. Indicates whether confidence intervals should be computed. Default=FALSE.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the difference in means
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

mixmodel_cont(data = trial_data, arm = 3, ci = TRUE)
```

---

piecewise_cal_cont	<i>Model-based analysis for continuous data using discontinuous piecewise polynomials per calendar time unit</i>
--------------------	--

---

**Description**

This function performs linear regression taking into account all trial data until the arm under study leaves the trial and adjusting for time using discontinuous piecewise polynomials in each calendar time unit.

**Usage**

```
piecewise_cal_cont(
  data,
  arm,
  alpha = 0.025,
  unit_size = 25,
  ncc = TRUE,
  poly_degree = 3,
  check = TRUE,
  ...
)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'j'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
unit_size	Integer. Number of patients per calendar time unit. Default=25.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
poly_degree	Integer. Degree of the piecewise polynomial. Default=3.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")
```

```
piecewise_cal_cont(data = trial_data, arm = 3)
```

---

piecewise_cont	<i>Model-based analysis for continuous data using discontinuous piecewise polynomials per period</i>
----------------	--

---

### Description

This function performs linear regression taking into account all trial data until the arm under study leaves the trial and adjusting for time using discontinuous piecewise polynomials in each period.

### Usage

```
piecewise_cont(
  data,
  arm,
  alpha = 0.025,
  ncc = TRUE,
  poly_degree = 3,
  check = TRUE,
  ...
)
```

### Arguments

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response', 'period' and 'j'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
poly_degree	Integer. Degree of the piecewise polynomial. Default=3.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

### Value

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

piecewise_cont(data = trial_data, arm = 3)
```

---

plot\_trial

*Function for visualizing the simulated trial*

---

**Description**

This function creates a plot visualizing the trial progress over time.

**Usage**

```
plot_trial(treatments)
```

**Arguments**

treatments      Vector with indices of assigned arms for each participant, ordered by time, e.g. column treatment from the dataframe resulting from the `datasim_bin()` or `datasim_cont()` function.

**Value**

ggplot showing trial progress over time.

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")

plot_trial(treatments = trial_data$treatment)
```

---

poolmodel\_bin                      *Pooled analysis for binary data*

---

### Description

This function performs pooled analysis (naively pooling concurrent and non-concurrent controls without adjustment) using a logistic model.

### Usage

```
poolmodel_bin(data, arm, alpha = 0.025, check = TRUE, ...)
```

### Arguments

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

### Details

The pooled analysis takes into account only the data from the evaluated experimental treatment arm and the whole control arm and uses a logistic regression model to evaluate the given treatment arm. Denoting by  $y_j$  the response probability for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$g(E(y_j)) = \eta_0 + \theta_M \cdot I(k_j = M)$$

where  $g(\cdot)$  denotes the logit link function and  $\eta_0$  is the log odds in the control arm;  $\theta_M$  represents the log odds ratio of treatment  $M$  and control.

### Value

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the log-odds ratio
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")
```

```
poolmodel_bin(data = trial_data, arm = 3)
```

---

poolmodel\_cont

*Pooled analysis for continuous data*

---

**Description**

This function performs pooled analysis (naively pooling concurrent and non-concurrent controls without adjustment) using a linear model.

**Usage**

```
poolmodel_cont(data, arm, alpha = 0.025, check = TRUE, ...)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The pooled analysis takes into account only the data from the evaluated experimental treatment arm and the whole control arm and uses a linear regression model to evaluate the given treatment arm. Denoting by  $y_j$  the continuous response for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$E(y_j) = \eta_0 + \theta_M \cdot I(k_j = M)$$

where  $\eta_0$  is the response in the control arm;  $\theta_M$  represents the treatment effect of treatment  $M$  as compared to control.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the difference in means
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

poolmodel_cont(data = trial_data, arm = 3)
```

---

seasonal_trend	<i>Generation of a seasonal trend</i>
----------------	---------------------------------------

---

**Description**

This function generates a time trend for given time points in the trial according to a periodic function.

**Usage**

```
seasonal_trend(j, lambda, n_wave, n_total)
```

**Arguments**

j	Time points for which the trend should be generated.
lambda	Strength of time trend.
n_wave	How many cycles (waves) should the time trend have ( $\psi$ ).
n_total	Total sample size in the trial.

**Details**

The time trend is generated according to the function  $f(j) = \lambda \cdot \sin(\psi \cdot 2\pi \cdot \frac{j-1}{N-1})$ , where  $N$  is the total sample size (parameter n\_total) and the parameter  $\psi$  corresponds to the input parameter n\_wave.

**Value**

Time trend for time points  $j$ .

**Author(s)**

Marta Bofill Roig, Pavla Krotka

---

sepmode<sub>l</sub>\_adj\_bin      *Separate analysis for binary data adjusted for periods*

---

**Description**

This function performs separate analysis (only taking into account concurrent controls) using a logistic model and adjusting for periods, if the treatment arm stays in the trial for more than one period.

**Usage**

```
sepmodel_adj_bin(data, arm, alpha = 0.025, check = TRUE, ...)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The adjusted separate analysis takes into account only the data from the evaluated experimental treatment arm and its concurrent controls and adjusts for the time effect by including the factor period (defined as a time interval bounded by any treatment arm entering or leaving the platform). The time is then modelled as a step-function with jumps at the beginning of each period. Denoting by  $y_j$  the response probability for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$g(E(y_j)) = \eta_0 + \theta_M \cdot I(k_j = M) + \sum_{s=S_{M_1}+1}^{S_{M_2}} \tau_s \cdot I(t_j \in T_{S_s})$$

where  $g(\cdot)$  denotes the logit link function and  $\eta_0$  is the log odds in the concurrent controls;  $\theta_M$  represents the log odds ratio of treatment  $M$  and control;  $\tau_s$  indicates the stepwise period effect in terms of the log odds ratio between periods  $S_{M_1}$  and  $s$  ( $s = S_{M_1} + 1, \dots, S_{M_2}$ ), where  $S_{M_1}$  and  $S_{M_2}$  denote the periods, in which the investigated treatment arm joined and left the trial, respectively.

If the data consists of only one period, the period is not used as covariate.

### Value

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the log-odds ratio
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

### Author(s)

Pavla Krotka

### Examples

```
trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")

sepmode1_adj_bin(data = trial_data, arm = 3)
```

---

sepmode1\_adj\_cont      *Separate analysis for continuous data adjusted for periods*

---

### Description

This function performs separate analysis (only taking into account concurrent controls) using a linear model and adjusting for periods, if the treatment arm stays in the trial for more than one period.

### Usage

```
sepmode1_adj_cont(data, arm, alpha = 0.025, check = TRUE, ...)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The adjusted separate analysis takes into account only the data from the evaluated experimental treatment arm and its concurrent controls and adjusts for the time effect by including the factor period (defined as a time interval bounded by any treatment arm entering or leaving the platform). The time is then modelled as a step-function with jumps at the beginning of each period. Denoting by  $y_j$  the response probability for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$E(y_j) = \eta_0 + \theta_M \cdot I(k_j = M) + \sum_{s=S_{M_1}+1}^{S_{M_2}} \tau_s \cdot I(t_j \in T_{S_s})$$

where  $\eta_0$  is the response in the concurrent controls;  $\theta_M$  represents the treatment effect of treatment  $M$  as compared to control;  $\tau_s$  indicates the stepwise period effect between periods  $S_{M_1}$  and  $s$  ( $s = S_{M_1} + 1, \dots, S_{M_2}$ ), where  $S_{M_1}$  and  $S_{M_2}$  denote the periods, in which the investigated treatment arm joined and left the trial, respectively.

If the data consists of only one period, the period is not used as covariate.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

sepmode_adj_cont(data = trial_data, arm = 3)
```

sepmode\_bin

*Separate analysis for binary data***Description**

This function performs separate analysis (only taking into account concurrent controls) using a logistic model.

**Usage**

```
sepmode_bin(data, arm, alpha = 0.025, check = TRUE, ...)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The separate analysis takes into account only the data from the evaluated experimental treatment arm and its concurrent controls and uses a logistic regression model to evaluate the given treatment arm. Denoting by  $y_j$  the response probability for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$g(E(y_j)) = \eta_0 + \theta_M \cdot I(k_j = M)$$

where  $g(\cdot)$  denotes the logit link function and  $\eta_0$  is the log odds in the concurrent controls;  $\theta_M$  represents the log odds ratio of treatment  $M$  and control.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- p-val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the log-odds ratio
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")

sepmode1_bin(data = trial_data, arm = 3)
```

---

sepmode1\_cont

*Separate analysis for continuous data*

---

**Description**

This function performs separate analysis (only taking into account concurrent controls) using a linear model.

**Usage**

```
sepmode1_cont(data, arm, alpha = 0.025, check = TRUE, ...)
```

**Arguments**

data	Data frame trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The separate analysis takes into account only the data from the evaluated experimental treatment arm and its concurrent controls and uses a linear regression model to evaluate the given treatment arm. Denoting by  $y_j$  the continuous response for patient  $j$ , by  $k_j$  the arm patient  $j$  was allocated to, and by  $M$  the treatment arm under evaluation, the regression model is given by:

$$E(y_j) = \eta_0 + \theta_M \cdot I(k_j = M)$$

where  $\eta_0$  is the response in the concurrent controls;  $\theta_M$  represents the treatment effect of treatment  $M$  as compared to control.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the difference in means
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- model - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

sepmode_cont(data = trial_data, arm = 3)
```

---

sim\_study

*Wrapper function performing simulation studies for a given set of scenarios (not parallelized)*

---

**Description**

This function performs a simulation study for a given set of scenarios, analyzing simulated data using different models as indicated by the user. Performs inference for indicated experimental treatment arms. Simulates the probability to reject  $H_0$  based on a given number of replications.

**Usage**

```
sim_study(
  nsim,
  scenarios,
  arms,
  models = c("fixmodel", "sepmode", "poolmodel"),
  endpoint,
  verbose = TRUE
)
```

**Arguments**

nsim	Integer. Number of replications. Must be larger than 1.
scenarios	Data frame containing all parameters for scenarios that should be simulated.
arms	Integer vector with treatment arms to perform inference on. These arms are compared to the control group. Default - all arms except the first one.
models	Character vector with models that should be used for the analysis. Default=c("fixmodel", "sepmode", "poolmodel"). Available models for continuous endpoints are: 'fixmodel', 'fixmodel_cal', 'gam', 'MAPprior', 'mixmodel', 'mixmodel_cal', 'mixmodel_AR1', 'mixmodel_AR1_cal', 'piecewise', 'piecewise_cal', 'poolmodel', 'sepmode', 'sepmode_adj', 'splines', 'splines_cal', 'timemachine'. Available models for binary endpoints are: 'fixmodel', 'fixmodel_cal', 'MAPprior', 'poolmodel', 'sepmode', 'sepmode_adj', 'timemachine'.
endpoint	Endpoint indicator. "cont" for continuous endpoints, "bin" for binary endpoints.
verbose	Logical. Indicates whether to print a message (system time and number of finished scenarios) after simulating each scenario in order to track the progress of the simulations. Default=TRUE.

**Value**

Data frame with all considered scenarios and corresponding results - the probability to reject  $H_0$ .

**Author(s)**

Pavla Krotka

**Examples**

```
# Create data frame with all parameters:
sim_scenarios <- data.frame(num_arms = 4,
  n_arm = 250,
  d1 = 250*0,
  d2 = 250*1,
  d3 = 250*2,
  d4 = 250*3,
  period_blocks = 2,
  mu0 = 0,
```

```

sigma = 1,
theta1 = 0,
theta2 = 0,
theta3 = 0,
theta4 = 0,
lambda0 = rep(seq(-0.15, 0.15, length.out = 9), 2),
lambda1 = rep(seq(-0.15, 0.15, length.out = 9), 2),
lambda2 = rep(seq(-0.15, 0.15, length.out = 9), 2),
lambda3 = rep(seq(-0.15, 0.15, length.out = 9), 2),
lambda4 = rep(seq(-0.15, 0.15, length.out = 9), 2),
trend = c(rep("linear", 9), rep("stepwise_2", 9)),
alpha = 0.025,
ncc = TRUE)

# Run simulation study:
sim_results <- sim_study(nsim = 100, scenarios = sim_scenarios, arms = c(3, 4),
models = c("fixmodel", "sepmode", "poolmodel"), endpoint = "cont")

```

---

sim_study_par	<i>Wrapper function performing simulation studies for a given set of scenarios (parallelized on replication level)</i>
---------------	--

---

## Description

This function performs a simulation study for a given set of scenarios, analyzing simulated data using different models as indicated by the user. Performs inference for indicated experimental treatment arms. Simulates the probability to reject  $H_0$ , and the bias, as well as the mean squared error (MSE) of the treatment effect estimates based on a given number of replications.

## Usage

```

sim_study_par(
  nsim,
  scenarios,
  arms,
  models = c("fixmodel", "sepmode", "poolmodel"),
  endpoint,
  perc_cores = 0.9,
  verbose = TRUE
)

```

## Arguments

nsim	Integer. Number of replications. Must be larger than 1.
scenarios	Data frame containing all parameters for scenarios that should be simulated.

arms	Integer vector with treatment arms to perform inference on. These arms are compared to the control group. Default - all arms except the first one.
models	Character vector with models that should be used for the analysis. Default=c("fixmodel", "sepmodel", "poolmodel"). Available models for continuous endpoints are: 'fixmodel', 'fixmodel_cal', 'gam', 'MAPprior', 'mixmodel', 'mixmodel_cal', 'mixmodel_AR1', 'mixmodel_AR1_cal', 'piecewise', 'piecewise_cal', 'poolmodel', 'sepmodel', 'sepmodel_adj', 'splines', 'splines_cal', 'timemachine'. Available models for binary endpoints are: 'fixmodel', 'fixmodel_cal', 'MAPprior', 'poolmodel', 'sepmodel', 'sepmodel_adj', 'timemachine'.
endpoint	Endpoint indicator. "cont" for continuous endpoints, "bin" for binary endpoints.
perc_cores	Double. What percentage of available cores should be used for the simulations. Default=0.9.
verbose	Logical. Indicates whether to print a message (system time and number of finished scenarios) after simulating each scenario in order to track the progress of the simulations. Default=TRUE.

### Value

Data frame with all considered scenarios and corresponding results - the probability to reject  $H_0$ , and the bias, as well as the mean squared error (MSE) of the treatment effect estimates.

### Author(s)

Pavla Krotka

### Examples

```
# Create data frame with all parameters:
sim_scenarios <- data.frame(num_arms = 4,
  n_arm = 250,
  d1 = 250*0,
  d2 = 250*1,
  d3 = 250*2,
  d4 = 250*3,
  period_blocks = 2,
  mu0 = 0,
  sigma = 1,
  theta1 = 0,
  theta2 = 0,
  theta3 = 0,
  theta4 = 0,
  lambda0 = rep(seq(-0.15, 0.15, length.out = 9), 2),
  lambda1 = rep(seq(-0.15, 0.15, length.out = 9), 2),
  lambda2 = rep(seq(-0.15, 0.15, length.out = 9), 2),
  lambda3 = rep(seq(-0.15, 0.15, length.out = 9), 2),
  lambda4 = rep(seq(-0.15, 0.15, length.out = 9), 2),
  trend = c(rep("linear", 9), rep("stepwise_2", 9)),
  alpha = 0.025,
  ncc = TRUE)
```

```
# Run simulation study:
sim_results <- sim_study_par(nsim = 100, scenarios = sim_scenarios, arms = c(3, 4),
models = c("fixmodel", "sepmodel", "poolmodel"), endpoint = "cont")
```

---

splines_cal_cont	<i>Spline regression analysis for continuous data with knots placed according to calendar time units</i>
------------------	--

---

### Description

This function performs linear regression taking into account all trial data until the arm under study leaves the trial and adjusting for time using regression splines with knots placed according to calendar time units.

### Usage

```
splines_cal_cont(
  data,
  arm,
  alpha = 0.025,
  unit_size = 25,
  ncc = TRUE,
  bs_degree = 3,
  check = TRUE,
  ...
)
```

### Arguments

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'j'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
unit_size	Integer. Number of patients per calendar time unit. Default=25.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
bs_degree	Integer. Degree of the polynomial spline. Default=3 for cubic spline.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- p\_val - p-value (one-sided)
- treat\_effect - estimated treatment effect in terms of the difference in means
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- knots - positions of the knots in terms of patient index
- model - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")

splines_cal_cont(data = trial_data, arm = 3)
```

---

splines_cont	<i>Spline regression analysis for continuous data with knots placed according to periods</i>
--------------	--

---

**Description**

This function performs linear regression taking into account all trial data until the arm under study leaves the trial and adjusting for time using regression splines with knots placed according to periods.

**Usage**

```
splines_cont(
  data,
  arm,
  alpha = 0.025,
  ncc = TRUE,
  bs_degree = 3,
  check = TRUE,
  ...
)
```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response', 'period' and 'j'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Significance level (one-sided). Default=0.025.
ncc	Logical. Indicates whether to include non-concurrent data into the analysis. Default=TRUE.
bs_degree	Integer. Degree of the polynomial spline. Default=3 for cubic spline.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Value**

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - p-value (one-sided)
- `treat_effect` - estimated treatment effect in terms of the difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  confidence interval
- `reject_h0` - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )
- `knots` - positions of the knots in terms of patient index
- `model` - fitted model

**Author(s)**

Pavla Krotka

**Examples**

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
  theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")
```

```
splines_cont(data = trial_data, arm = 3)
```

---

sw_trend	<i>Generation of stepwise trend with equal jumps between periods</i>
----------	--

---

**Description**

This function generates a stepwise trend for a given period. No time trend is assumed in the first period.

**Usage**

```
sw_trend(cj, lambda)
```

**Arguments**

cj	Period indicator.
lambda	Strength of time trend.

**Details**

The time trend is generated according to the function  $f(j) = \lambda \cdot (c_j - 1)$ , where  $c_j$  is an index of the period patient  $j$  was enrolled in.

**Value**

Time trend in period  $c_j$ .

**Author(s)**

Marta Bofill Roig, Pavla Krotka

---

timemachine_bin	<i>Time machine analysis for binary data</i>
-----------------	--

---

**Description**

This function performs analysis of binary data using the Time Machine approach. It takes into account all data until the investigated arm leaves the trial. It is based on logistic regression with treatment as a categorical variable and covariate adjustment for time via a second-order Bayesian normal dynamic linear model (separating the trial into buckets of pre-defined size).

**Usage**

```

timemachine_bin(
  data,
  arm,
  alpha = 0.025,
  prec_theta = 0.001,
  prec_eta = 0.001,
  tau_a = 0.1,
  tau_b = 0.01,
  bucket_size = 25,
  check = TRUE,
  ...
)

```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_bin()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Decision boundary (one-sided). Default=0.025.
prec_theta	Double. Precision ( $1/\sigma_\theta^2$ ) of the prior regarding the treatment effect $\theta$ . I.e. $\theta \sim N(0, \sigma_\theta^2)$ . Default=0.001.
prec_eta	Double. Precision ( $1/\sigma_{\eta_0}^2$ ) of the prior regarding the control log-odds $\eta_0$ . I.e. $\eta_0 \sim N(0, \sigma_{\eta_0}^2)$ . Default=0.001.
tau_a	Double. Parameter $a_\tau$ of the Gamma distribution for the precision parameter $\tau$ in the model for the time trend. I.e., $\tau \sim \text{Gamma}(a_\tau, b_\tau)$ . Default=0.1.
tau_b	Double. Parameter $b_\tau$ of the Gamma distribution for the precision parameter $\tau$ in the model for the time trend. I.e., $\tau \sim \text{Gamma}(a_\tau, b_\tau)$ . Default=0.01.
bucket_size	Integer. Number of patients per time bucket. Default=25.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

**Details**

The Time Machine divides the trial duration into  $C$  calendar time intervals of equal length ("buckets"), which are indexed backwards in time. That is to say, the most recent time interval is denoted by  $c = 1$  and the time interval corresponding to the beginning of the trial by  $c = C$ . The analysis is performed as soon as the analyzed treatment arm finishes in the trial.

The model is defined as follows:

$$g(E(y_j)) = \eta_0 + \theta_{k_j} + \alpha_{c_j}$$

where  $y_j$  is the binary response for patient  $j$  and  $g(\cdot)$  is the logit link function, which maps the expected value of the patient response to the linear predictors in the model. The model intercept  $\eta_0$  denotes the response of the control group at time of the analysis,  $\theta_{k_j}$  is the effect of the treatment arm  $k$  that patient  $j$  was enrolled in, relative to control in terms of the log odds ratio. For the parameters  $\eta_0$  and  $\theta_{k_j}$ , normal prior distributions are assumed, with mean 0 and variances  $\sigma_{\eta_0}^2$  and  $\sigma_{\theta}^2$ , respectively:

$$\eta_0 \sim \mathcal{N}(0, \sigma_{\eta_0}^2)$$

$$\theta_{k_j} \sim \mathcal{N}(0, \sigma_{\theta}^2)$$

In the Time Machine, time effect is represented by  $\alpha_{c_j}$ , which is the change in the response in time bucket  $c_j$  (which denotes the time bucket in which patient  $j$  is enrolled) compared to the most recent time bucket  $c = 1$  and is modeled using a Bayesian second-order normal dynamic linear model. This creates a smoothing over the control response, such that closer time buckets are modeled with more similar response rates:

$$\alpha_1 = 0$$

$$\alpha_2 \sim \mathcal{N}(0, 1/\tau)$$

$$\alpha_c \sim \mathcal{N}(2\alpha_{c-1} - \alpha_{c-2}, 1/\tau), 3 \leq c \leq C$$

where  $\tau$  denotes the drift parameter that controls the degree of smoothing over the time buckets and is assumed to have a Gamma hyperprior distribution:

$$\tau \sim \text{Gamma}(a_\tau, b_\tau)$$

## Value

List containing the following elements regarding the results of comparing arm to control:

- p\_val - posterior probability that the log-odds ratio is less than zero
- treat\_effect - posterior mean of log-odds ratio
- lower\_ci - lower limit of the  $(1-2*\alpha)*100\%$  credible interval for log-odds ratio
- upper\_ci - upper limit of the  $(1-2*\alpha)*100\%$  credible interval for log-odds ratio
- reject\_h0 - indicator of whether the null hypothesis was rejected or not ( $p\_val < \alpha$ )

## Author(s)

Dominic Magirr, Peter Jacko

## References

The Bayesian Time Machine: Accounting for Temporal Drift in Multi-arm Platform Trials. Saville, B. R., Berry, D. A., et al. *Clinical Trials* 19.5 (2022): 490-501.

**Examples**

```

trial_data <- datasim_bin(num_arms = 3, n_arm = 100, d = c(0, 100, 250),
p0 = 0.7, OR = rep(1.8, 3), lambda = rep(0.15, 4), trend="stepwise")

timemachine_bin(data = trial_data, arm = 3)

```

---

timemachine\_cont      *Time machine analysis for continuous data*

---

**Description**

This function performs analysis of continuous data using the Time Machine approach. It takes into account all data until the investigated arm leaves the trial. It is based on linear regression with treatment as a categorical variable and covariate adjustment for time via a second-order Bayesian normal dynamic linear model (separating the trial into buckets of pre-defined size).

**Usage**

```

timemachine_cont(
  data,
  arm,
  alpha = 0.025,
  prec_theta = 0.001,
  prec_eta = 0.001,
  tau_a = 0.1,
  tau_b = 0.01,
  prec_a = 0.001,
  prec_b = 0.001,
  bucket_size = 25,
  check = TRUE,
  ...
)

```

**Arguments**

data	Data frame with trial data, e.g. result from the <code>datasim_cont()</code> function. Must contain columns named 'treatment', 'response' and 'period'.
arm	Integer. Index of the treatment arm under study to perform inference on (vector of length 1). This arm is compared to the control group.
alpha	Double. Decision boundary (one-sided). Default=0.025.
prec_theta	Double. Precision ( $1/\sigma_\theta^2$ ) of the prior regarding the treatment effect $\theta$ . I.e. $\theta \sim N(0, \sigma_\theta^2)$ . Default=0.001.
prec_eta	Double. Precision ( $1/\sigma_{\eta_0}^2$ ) of the prior regarding the control mean $\eta_0$ . I.e. $\eta_0 \sim N(0, \sigma_{\eta_0}^2)$ . Default=0.001.

tau_a	Double. Parameter $a_\tau$ of the Gamma distribution for the precision parameter $\tau$ in the model for the time trend. I.e., $\tau \sim \text{Gamma}(a_\tau, b_\tau)$ . Default=0.1.
tau_b	Double. Parameter $b_\tau$ of the Gamma distribution for the precision parameter $\tau$ in the model for the time trend. I.e., $\tau \sim \text{Gamma}(a_\tau, b_\tau)$ . Default=0.01.
prec_a	Double. Parameter $a_{\sigma^2}$ of the Gamma distribution regarding the precision of the responses. I.e., $1/\sigma^2 \sim \text{Gamma}(a_{\sigma^2}, b_{\sigma^2})$ . Default=0.001.
prec_b	Double. Parameter $b_{\sigma^2}$ of the Gamma distribution regarding the precision of the responses. I.e., $1/\sigma^2 \sim \text{Gamma}(a_{\sigma^2}, b_{\sigma^2})$ . Default=0.001.
bucket_size	Integer. Number of patients per time bucket. Default=25.
check	Logical. Indicates whether the input parameters should be checked by the function. Default=TRUE, unless the function is called by a simulation function, where the default is FALSE.
...	Further arguments passed by wrapper functions when running simulations.

### Details

The Time Machine divides the trial duration into  $C$  calendar time intervals of equal length ("buckets"), which are indexed backwards in time. That is to say, the most recent time interval is denoted by  $c = 1$  and the time interval corresponding to the beginning of the trial by  $c = C$ . The analysis is performed as soon as the analyzed treatment arm finishes in the trial.

The model is defined as follows:

$$E(y_j) = \eta_0 + \theta_{k_j} + \alpha_{c_j}$$

where  $y_j$  is the continuous response for patient  $j$ . The model intercept  $\eta_0$  denotes the response of the control group at time of the analysis,  $\theta_{k_j}$  is the effect of the treatment arm  $k$  that patient  $j$  was enrolled in, relative to control. For the parameters  $\eta_0$  and  $\theta_{k_j}$ , normal prior distributions are assumed, with mean 0 and variances  $\sigma_{\eta_0}^2$  and  $\sigma_{\theta}^2$ , respectively:

$$\eta_0 \sim \mathcal{N}(0, \sigma_{\eta_0}^2)$$

$$\theta_{k_j} \sim \mathcal{N}(0, \sigma_{\theta}^2)$$

In the Time Machine, time effect is represented by  $\alpha_{c_j}$ , which is the change in the response in time bucket  $c_j$  (which denotes the time bucket in which patient  $j$  is enrolled) compared to the most recent time bucket  $c = 1$  and is modeled using a Bayesian second-order normal dynamic linear model. This creates a smoothing over the control response, such that closer time buckets are modeled with more similar response rates:

$$\alpha_1 = 0$$

$$\alpha_2 \sim \mathcal{N}(0, 1/\tau)$$

$$\alpha_c \sim \mathcal{N}(2\alpha_{c-1} - \alpha_{c-2}, 1/\tau), 3 \leq c \leq C$$

where  $\tau$  denotes the drift parameter that controls the degree of smoothing over the time buckets and is assumed to have a Gamma hyperprior distribution:

$$\tau \sim \text{Gamma}(a_\tau, b_\tau)$$

The precision of the individual patient responses ( $1/\sigma^2$ ) is also assumed to have a Gamma hyper-prior distribution:

$$1/\sigma^2 \sim \text{Gamma}(a_{\sigma^2}, b_{\sigma^2})$$

### Value

List containing the following elements regarding the results of comparing arm to control:

- `p_val` - posterior probability that the difference in means is less than zero
- `treat_effect` - posterior mean of difference in means
- `lower_ci` - lower limit of the  $(1-2*\alpha)*100\%$  credible interval for difference in means
- `upper_ci` - upper limit of the  $(1-2*\alpha)*100\%$  credible interval for difference in means
- `reject_h0` - indicator of whether the null hypothesis was rejected or not (`p_val < alpha`)

### Author(s)

Dominic Magirr, Peter Jacko

### Examples

```
trial_data <- datasim_cont(num_arms = 3, n_arm = 100, d = c(0, 100, 250),  
theta = rep(0.25, 3), lambda = rep(0.15, 4), sigma = 1, trend = "linear")
```

```
timemachine_cont(data = trial_data, arm = 3)
```

# Index

[datasim\\_bin](#), 2  
[datasim\\_cont](#), 6

[fixmodel\\_bin](#), 10  
[fixmodel\\_cal\\_bin](#), 11  
[fixmodel\\_cal\\_cont](#), 13  
[fixmodel\\_cont](#), 15

[gam\\_cont](#), 16  
[get\\_ss\\_matrix](#), 18

[inv\\_u\\_trend](#), 19

[linear\\_trend](#), 19

[MAPprior\\_bin](#), 20  
[MAPprior\\_cont](#), 23  
[mixmodel\\_AR1\\_cal\\_cont](#), 26  
[mixmodel\\_AR1\\_cont](#), 27  
[mixmodel\\_cal\\_cont](#), 28  
[mixmodel\\_cont](#), 30

[piecewise\\_cal\\_cont](#), 31  
[piecewise\\_cont](#), 33  
[plot\\_trial](#), 34  
[poolmodel\\_bin](#), 35  
[poolmodel\\_cont](#), 36

[seasonal\\_trend](#), 37  
[sepmodel\\_adj\\_bin](#), 38  
[sepmodel\\_adj\\_cont](#), 39  
[sepmodel\\_bin](#), 41  
[sepmodel\\_cont](#), 42  
[sim\\_study](#), 43  
[sim\\_study\\_par](#), 45  
[splines\\_cal\\_cont](#), 47  
[splines\\_cont](#), 48  
[sw\\_trend](#), 50

[timemachine\\_bin](#), 50  
[timemachine\\_cont](#), 53