Package 'trialr'

April 21, 2019

Version 0.1.0

Date 2019-04-2117

Title Clinical Trial Designs in 'RStan'

Description A showcase of Bayesian clinical trial designs, implemented in 'RStan' and R. Some designs are implemented in R for the first time (e.g. 'EffTox' by Thall and Cook, 2004, <https://biostatistics.mdanderson.org/softwaredownload/SingleSoftware.aspx?Software_Id=2>). Given the emphasis on flexibility in Bayesian analysis, the implementations in a common language and style may serve as a cookbook to encourage the use of Bayesian methods in trials. Compiled 'RStan' models are provided in addition to helper classes and functions to perform simulations and inference on observed trial outcomes. There is a preponderance of early phase trial designs because this is where Bayesian methods are used most. If there is a published Bayesian design you want implemented in 'Stan', get in touch.

Maintainer Kristian Brock <kristian.brock@gmail.com>

License GPL (>= 3)

Encoding UTF-8

LazyData true

ByteCompile true

Depends R (>= 3.4.0), methods,Rcpp (>= 1.0.1)

Imports rstan (>= 2.18.2), rstantools (>= 1.5.1), loo (>= 1.1.0), dplyr, tidyr, magrittr, stringr, ggplot2, gtools

LinkingTo StanHeaders (>= 2.18.1), rstan (>= 2.18.2), BH (>= 1.69.0-1), Rcpp (>= 1.0.1), RcppEigen (>= 0.3.3.5.0)

SystemRequirements GNU make

NeedsCompilation yes

RoxygenNote 6.1.1

VignetteBuilder knitr

URL https://github.com/brockk/trialr

BugReports https://github.com/brockk/trialr/issues

Suggests testthat, knitr, rmarkdown, ggridges, covr

Author Kristian Brock [aut, cre] (<https://orcid.org/0000-0002-3921-0166>),
Trustees of Columbia University [cph]

Repository CRAN

Date/Publication 2019-04-21 21:00:03 UTC

R topics documented:

trialr-package *The 'trialr' package.*

Description

trialr collects in one place Bayesian clinical trial designs and methods. Models are implemented in Stan and helper functions are provided in R.

References

Stan Development Team (2018). RStan: the R interface to Stan. R package version 2.18.2. http://mc-stan.org

as.data.frame.crm_fit *Convert crm_fit object to* data.frame.

Description

Convert crm_fit object to `data.frame`.

Usage

```
## S3 method for class 'crm_fit'
as.data.frame(x, ...)
```

Arguments

x	crm_fit object to convert.
...	Extra parameters, passed onwards.

Value

A `data.frame`

as.data.frame.efftox_fit

Convert efftox_fit object to `data.frame`.

Description

Convert efftox_fit object to `data.frame`.

Usage

```
## S3 method for class 'efftox_fit'
as.data.frame(x, ...)
```

Arguments

`x`	`efftox_fit` object to convert.
`...`	Extra parameters, passed onwards.

Value

A `data.frame`

crm_fit-class

Class of model fit by **trialr** *using the CRM dose-finding design.*

Description

Class of model fit by **trialr** using the CRM dose-finding design.

Usage

```
crm_fit(dose_indices, recommended_dose, prob_tox, median_prob_tox,
  modal_mtd_candidate, prob_mtd, dat, fit)
```

Arguments

`dose_indices`	A vector of integers representing the dose-levels under consideration.
`recommended_dose`	An integer representing the dose-level recommended for the next patient or cohort; or `NA` if stopping is recommended. The recommended dose typically has associated probability of DLT closest to the target toxicity rate. Contrast to `modal_mtd_candidate`.
`prob_tox`	The posterior mean probabilities of toxicity at doses 1:n; a vector of numbers between 0 and 1.

median_prob_tox
: The posterior median probabilities of toxicity at doses 1:n; a vector of numbers between 0 and 1.

modal_mtd_candidate
: An integer representing the dose-level most likely to be the MTD, i.e. the dose-level that maximises prob_mtd.

prob_mtd
: The posterior probability that each dose is the MTD, by the chosen model; a vector of numbers between 0 and 1.

dat
: Object crm_params containing data passed to sampling.

fit
: An object of class stanfit, containing the posterior samples.

Details

See methods(class = "crm_fit") for an overview of available methods.

See Also

stan_crm

---

crm_params-class *Container class for parameters to fit the CRM models in trialr.*

Description

Container class for parameters to fit the CRM models in trialr.

Usage

```
crm_params(skeleton, target, a0 = NULL, alpha_mean = NULL,
  alpha_sd = NULL, beta_mean = NULL, beta_sd = NULL, beta_shape = NULL,
  beta_inverse_scale = NULL)
```

Arguments

skeleton
: a vector of the prior guesses of toxicity at doses. This should be a monotonically-increasing vector of numbers between 0 and 1.

target
: the target toxicity probability, a number between 0 and 1. This value would normally be one of the values in skeleton, but that is not a requirement.

a0
: Value of fixed intercept parameter. Only required for certain models. See Details.

alpha_mean
: Prior mean of intercept variable for normal prior. Only required for certain models. See Details.

alpha_sd
: Prior standard deviation of intercept variable for normal prior. Only required for certain models. See Details.

beta_mean
: Prior mean of gradient variable for normal prior. Only required for certain models. See Details.

`beta_sd` Prior standard deviation of slope variable for normal prior. Only required for certain models. See Details.

`beta_shape` Prior shape parameter of slope variable for gamma prior. Only required for certain models. See Details.

`beta_inverse_scale` Prior inverse scale parameter of slope variable for gamma prior. Only required for certain models. See Details.

Details

Different model parameterisations require that difference parameter values are specified.

Requirements of `empiric` model

* beta_sd

Requirements of `logistic` model

* a0 * beta_mean * beta_sd

Requirements of `logistic_gamma` model

* a0 * beta_shape * beta_inverse_scale

Requirements of `logistics` model

* a0 * alpha_mean * alpha_sd * beta_mean * beta_sd

See Also

`stan_crm`

`crm_process` *Process RStan samples from a CRM model*

Description

Internal function to process rstan samples from a CRM model to make inferences about dose-toxicity and which dose should be recommended next. Typically, this function is not required to be called explicitly by the user because `stan_crm` will call it implicitly.

Usage

```
crm_process(dat, fit)
```

Arguments

dat	An instance of crm_params, a list of CRM parameters.
fit	An instance of rstan::stanmodel, derived by fitting one of the trialr CRM models.

Value

An instance of crm_fit.

---

df_parse_outcomes *Parse a string of dose-finding trial outcomes to binary vector notation.*

Description

Parse a string of dose-finding trial outcomes to the binary vector notation required by Stan for model invocation. The outcome string describes the doses given and outcomes observed. The format of the string is the pure phase I analogue to that described in Brock et al. (2017). The letters T and N are used to represents patients that experienced (T)oxicity and (N)o toxicity. These letters are concatenated after numerical dose-levels to convey the outcomes of cohorts of patients. For instance, 2NNT represents a cohort of three patients that were treated at dose-level 2, one of whom experienced toxicity, and two that did not. The results of cohorts are separated by spaces. Thus, 2NNT 1NN extends our previous example, where the next cohort of two were treated at dose-level 1 and neither experienced toxicity. See examples.

Usage

```
df_parse_outcomes(outcome_string, as.list = TRUE)
```

Arguments

outcome_string	character string, conveying doses given and outcomes observed.
as.list	TRUE (be default) to return a list; FALSE to return a data.frame

Value

If as.list == TRUE, a list with elements tox, doses and num_patients. These elements are congruent with those of the same name in crm_params, for example. If as.list == FALSE, a data.frame with columns tox and doses.

References

Brock, K., Billingham, L., Copland, M., Siddique, S., Sirovica, M., & Yap, C. (2017). Implementing the EffTox dose-finding design in the Matchpoint trial. BMC Medical Research Methodology, 17(1), 112. https://doi.org/10.1186/s12874-017-0381-x

Examples

```
x = df_parse_outcomes('1NNN 2NTN 3TTT')
x$num_patients
x$tox
sum(x$tox)
```

`efftox_analysis_to_df` *EffTox analysis to data.frame*

Description

Convenient function to turn an `efftox_fit` into a `data.frame`.

Usage

```
efftox_analysis_to_df(x)
```

Arguments

x	An instance of `efftox_fit`

Value

a `data.frame`

See Also

`stan_efftox`

Examples

```
fit <- stan_efftox_demo(outcome_str = '1N 2E 3B')
df <- efftox_analysis_to_df(fit)
df
```

`efftox_contour_plot` *Plot EffTox utility contours*

Description

Plot EffTox utility contours. The probability of efficacy is on the x-axis and toxicity on the y-axis. The zero-utility curve is plotted bolder. The three "hinge points" are plotted as blue triangles. Optional Prob(Efficacy) vs Prob(Toxicity) points can be added; these are shown as red numerals, enumerated in the order provided.

Usage

```
efftox_contour_plot(fit, use_ggplot = FALSE, prob_eff = fit$prob_eff,
  prob_tox = fit$prob_tox, num_points = 1000, util_vals = seq(-3, 3, by =
  0.2))
```

Arguments

`fit`	An instance of `efftox_fit`.
`use_ggplot`	logical, TRUE to use ggplot2. Defaults to FALSE to use standard R graphics.
`prob_eff`	vector of numbers between 0 and 1, containing the efficacy probabilities of extra points to add to the plot as points, e.g. the posterior mean efficacy probabilities of the doses under investigation. Paired with prob_tox, thus they should be the same length. Defaults to the values fitted by the model. Use NULL to supress.
`prob_tox`	vector of numbers between 0 and 1, containing the toxicity probabilities of extra points to add to the plot as points, e.g. the posterior mean toxicity probabilities of the doses under investigation. Paired with prob_eff, thus they should be the same length. Defaults to the values fitted by the model. Use NULL to supress.
`num_points`	integer for number of points to calculate on each curve. The default is 1000 and this should be plenty.
`util_vals`	A contour is plotted for each of these utility values. The default is contours spaced by 0.2 between from -3 and 3, i.e. `seq(-3, 3, by = 0.2)`.

Value

if `use_ggplot = TRUE`, an instance of `ggplot`; else no object is returned. Omit assignment in either case to just view the plot.

See Also

`stan_efftox`

Examples

```
fit <- stan_efftox_demo(outcome_str = '1N 2E 3B')
efftox_contour_plot(fit)
title('EffTox utility contours')
# The same with ggplot2
efftox_contour_plot(fit, use_ggplot = TRUE) +
                    ggplot2::ggtitle('EffTox utility contours')
```

`efftox_dtps` *Calculate dose-transition pathways for an EffTox study*

Description

Calculate dose-transition pathways for an EffTox study.

Usage

```
efftox_dtps(dat, cohort_sizes, next_dose, ...)
```

Arguments

`dat`	An instance of `efftox_params`, a list of EffTox parameters. An example is yielded by `efftox_parameters_demo`.
`cohort_sizes`	vector of future cohort sizes, i.e. positive integers. E.g. Tot calculate paths for the the next cohort of two followed by the next cohort of three, use `c(2, 3)`.
`next_dose`	the dose-level to be given to the immediately next cohort.
`...`	extra params passed to `rstan::sampling`.

Value

dose pathways in a `data.frame`.

References

Brock et al. (submitted 2017), Implementing the EffTox Dose-Finding Design in the Matchpoint Trial.

See Also

`efftox_params`

`efftox_parameters_demo`

Examples

```
# Calculate the paths for the first cohort of 3 in Thall et al 2014 example
dat <- efftox_parameters_demo()
## Not run:
dtps1 <- efftox_dtps(dat = dat, cohort_sizes = c(3), next_dose = 1)

## End(Not run)
# To calculate future paths in a partially-observed trial
dat <- efftox_parameters_demo()
dat$doses = array(c(1,1,1))
dat$eff = array(c(0,0,0))
dat$tox = array(c(1,1,1))
dat$num_patients = 3
## Not run:
dtps2 <- efftox_dtps(dat = dat, cohort_sizes = c(3), next_dose = 1)

## End(Not run)
```

---

efftox_fit-class *Class of model fit by* **trialr** *using the EffTox dose-finding design.*

Description

Class of model fit by **trialr** using the EffTox dose-finding design.

Usage

```
efftox_fit(dose_indices, recommended_dose, prob_eff, prob_tox, prob_acc_eff,
  prob_acc_tox, utility, post_utility, acceptable, dat, fit)
```

Arguments

dose_indices	A vector of integers representing the dose-levels under consideration.
recommended_dose	An integer representing the dose-level recommended for the next patient or cohort; or NA if stopping is recommended.
prob_eff	The posterior mean probabilities of efficacy at doses 1:n; a vector of numbers between 0 and 1.
prob_tox	The posterior mean probabilities of toxicity at doses 1:n; a vector of numbers between 0 and 1.
prob_acc_eff	The posterior mean probabilities that efficacy at the doses is acceptable, i.e. that it exceeds the minimum acceptable efficacy threshold; a vector of numbers between 0 and 1.
prob_acc_tox	The posterior mean probabilities that toxicity at the doses is acceptable, i.e. that it is less than the maximum toxicity threshold; a vector of numbers between 0 and 1.

utility	The utilities of doses 1:n, calculated by plugging the posterior mean probabilities of efficacy and toxicity into the utility formula, as advocated by Thall & Cook. Contrast to post_utility; a vector of numbers.
post_utility	The posterior mean utilities of doses 1:n, calculated from the posterior distributions of the utilities. This is in contrast to utility, which uses plug-in posterior means of efficacy and toxicity, as advocated by Thall & Cook; a vector of numbers.
acceptable	A vector of logical values to indicate whether doses 1:n are acceptable, according to the rules for acceptable efficacy & toxicity, and rules on not skipping untested doses.
dat	Object efftox_params containing data passed to sampling.
fit	An object of class stanfit, containing the posterior samples.

Details

See methods(class = "efftox_fit") for an overview of available methods.

See Also

stan_efftox stan_efftox_demo

---

efftox_get_tox *Get the Prob(Tox) for Prob(Eff) and utility pairs*

Description

Get the probability of toxicity for probability-of-efficacy and utility pairs

Usage

```
efftox_get_tox(eff, util, p, eff0, tox1)
```

Arguments

eff	Probability of efficacy; number between 0 and 1
util	Utility score; number
p	p-index of EffTox utility contours. Use efftox_solve_p
eff0	Efficacy probability required when toxicity is impossible; a number between 0 and 1
tox1	Toxicity probability permitted when efficacy is guaranteed; a number between 0 and 1

Value

Probability(s) of toxicity

Note

Various ways of vectorising the function are demonstrated in the examples

See Also

efftox_solve_p

Examples

```
p <- efftox_solve_p(0.5, 0.65, 0.7, 0.25)

prob_tox <- efftox_get_tox(0.7, 0, p, eff0 = 0.5, tox1 = 0.65)
round(prob_tox, 2) == 0.25

prob_tox <- efftox_get_tox(0.7, seq(-0.5, 0.25, by = 0.25), p, eff0 = 0.5,
                           tox1 = 0.65)
round(prob_tox, 2) == c(0.57, 0.41, 0.25, 0.09)

prob_tox <- efftox_get_tox(c(0.5, 0.7, 0.8), 0.25, p, eff0 = 0.5, tox1 = 0.65)
round(prob_tox, 2) == c(NaN, 0.09, 0.22)

prob_tox <- efftox_get_tox(c(0.5, 0.7, 0.8), c(-1, 0, 1), p, eff0 = 0.5,
                           tox1 = 0.65)
round(prob_tox, 2) == c(0.63, 0.25, NaN)
```

---

efftox_parameters_demo

Get parameters to run the EffTox demo

Description

Get parameters to run the EffTox demo. These match those used to demonstrate EffTox in Thall et al. 2014.

Usage

```
efftox_parameters_demo()
```

Value

a `list` of parameters, described in `efftox_params`

References

Thall, Herrick, Nguyen, Venier & Norris. 2014, Effective sample size for computing prior hyper-parameters in Bayesian phase I-II dose-finding

See Also

efftox_params

Examples

```
dat <- efftox_parameters_demo()
names(dat)
dat$real_doses == c(1, 2, 4, 6.6, 10)
```

---

efftox_params-class *Container class for parameters to fit the EffTox model in trialr.*

Description

Container class for parameters to fit the EffTox model in trialr.

Usage

```
efftox_params(real_doses, efficacy_hurdle, toxicity_hurdle, p_e, p_t, eff0,
  tox1, eff_star, tox_star, alpha_mean, alpha_sd, beta_mean, beta_sd,
  gamma_mean, gamma_sd, zeta_mean, zeta_sd, eta_mean, eta_sd, psi_mean, psi_sd)
```

Arguments

real_doses	a vector of numbers.The doses under investigation. They should be ordered from lowest to highest and be in consistent units. E.g., to conduct a dose-finding trial of doses 10mg, 20mg and 50mg, use c(10, 20, 50).
efficacy_hurdle	Minimum acceptable efficacy probability. A number between 0 and 1.
toxicity_hurdle	Maximum acceptable toxicity probability. A number between 0 and 1.
p_e	Certainty required to infer a dose is acceptable with regards to being probably efficacious; a number between 0 and 1.
p_t	Certainty required to infer a dose is acceptable with regards to being probably tolerable; a number between 0 and 1.
eff0	Efficacy probability required when toxicity is impossible; a number between 0 and 1 (see Details).
tox1	Toxicity probability permitted when efficacy is guaranteed; a number between 0 and 1 (see Details).
eff_star	Efficacy probability of an equi-utility third point (see Details).
tox_star	Toxicity probability of an equi-utility third point (see Details).
alpha_mean	The prior normal mean of the intercept term in the toxicity logit model. A number.

alpha_sd	The prior normal standard deviation of the intercept term in the toxicity logit model. A number.
beta_mean	The prior normal mean of the slope term in the toxicity logit model. A number.
beta_sd	The prior normal standard deviation of the slope term in the toxicity logit model. A number.
gamma_mean	The prior normal mean of the intercept term in the efficacy logit model. A number.
gamma_sd	The prior normal standard deviation of the intercept term in the efficacy logit model. A number.
zeta_mean	The prior normal mean of the slope term in the efficacy logit model. A number.
zeta_sd	The prior normal standard deviation of the slope term in the efficacy logit model. A number.
eta_mean	The prior normal mean of the squared term coefficient in the efficacy logit model. A number.
eta_sd	The prior normal standard deviation of the squared term coefficient in the efficacy logit model. A number.
psi_mean	The prior normal mean of the association term in the combined efficacy-toxicity model. A number.
psi_sd	The prior normal standard deviation of the association term in the combined efficacy-toxicity model. A number.

See Also

stan_efftox stan_efftox_demo

---

efftox_parse_outcomes *Parse a string of EffTox outcomes to binary vector notation.*

Description

Parse a string of EffTox outcomes to the binary vector notation required by Stan for model invocation. The outcome string describes the doses given and outcomes observed. The format of the string is described in Brock et al. (2017). The letters E, T, N and B are used to represents patients that experienced (E)fficacy only, (T)oxicity only, (B)oth efficacy and toxicity, and (N)either. These letters are concatenated after numerical dose-levels to convey the outcomes of cohorts of patients. For instance, `2ETB` represents a cohort of three patients that were treated at dose-level 2, and experienced efficacy, toxicity and both events, respectively. The results of cohorts are separated by spaces. Thus, `2ETB 1NN` extends our previous example, where the next cohort of two were treated at dose-level 1 and both patients experienced neither efficacy nor toxicity. See examples.

We present the notation in the EffTox setting but it is applicable in general seamless phase I/II dose-finding scenarios.

Usage

```
efftox_parse_outcomes(outcome_string, as.list = TRUE)
```

Arguments

outcome_string	character string, conveying doses given and outcomes observed.
as.list	TRUE (be default) to return a list; FALSE to return a data.frame

Value

If as.list == TRUE, a list with elements eff, tox, doses and num_patients. These elements are congruent with those of the same name in efftox_params. If as.list == FALSE, a data.frame with columns eff, tox, and doses.

References

Brock, K., Billingham, L., Copland, M., Siddique, S., Sirovica, M., & Yap, C. (2017). Implementing the EffTox dose-finding design in the Matchpoint trial. BMC Medical Research Methodology, 17(1), 112. https://doi.org/10.1186/s12874-017-0381-x

Examples

```
x = efftox_parse_outcomes('1NNE 2EEN 3TBB')
x$num_patients == 9
x$eff == c(0, 0, 1, 1, 1, 0, 0, 1, 1)
sum(x$tox) == 3
```

---

efftox_process *Process RStan samples from an EffTox model*

Description

Internal function to process rstan samples from an EffTox model to make inferences about dose-acceptability, dose-utility and which dose should be recommended next.

Usage

```
efftox_process(dat, fit)
```

Arguments

dat	An instance of efftox_params, a list of EffTox parameters. An example is yielded by efftox_parameters_demo.
fit	An instance of rstan::stanmodel, derived by fitting the trialr EffTox model.

Value

An instance of efftox_fit.

`efftox_simulate` *Run EffTox simulations*

Description

Run EffTox simulations for assumed true efficacy and toxicity curves.

Usage

```
efftox_simulate(dat, num_sims, first_dose, true_eff, true_tox, cohort_sizes,
  ...)
```

Arguments

`dat`	An instance of `efftox_params`, a list of EffTox parameters. An example is yielded by `efftox_parameters_demo`.
`num_sims`	integer, number of simulated iterations
`first_dose`	integer, the dose-level to give to patient 1, e.g. 1 for the lowest dose.
`true_eff`	the true probabilities of efficacy at the doses under investigation; a vector of numbers between 0 and 1.
`true_tox`	the true probabilities of toxicity at the doses under investigation; a vector of numbers between 0 and 1.
`cohort_sizes`	a vector of integer cohort sizes. A dose decision is made when each cohort is completed and the next cohort is treated at the recommended dose. To conduct a trial using at most 20 patients, where dose is re-evaluated after every second patient, use `rep(2, 10)`. To conduct a trial of 8 patients where dose is re-evaluated after each single patient, use `rep(1, 8)`. Cohort size need not be uniform. E.g. `c(rep(1, 5), rep(3, 10))` represents a trial where the dose is re-evaluated after each patient for the first 5 patients, and then after every third patient for a further 30 patients.
`...`	Extra parameters provided via the ellipsis are passed to `stan::sampling`

Value

A list with named elements `recommended_dose`, `efficacies`, `toxicities`, and `doses_given`.

Examples

```
dat <- efftox_parameters_demo()
set.seed(123)
# Let's say we want to use only 2 chains. Extra args are passed to stan
## Not run:
sims <- efftox_simulate(dat, num_sims = 10, first_dose = 1,
                        true_eff = c(0.20, 0.40, 0.60, 0.80, 0.90),
                        true_tox = c(0.05, 0.10, 0.15, 0.20, 0.40),
                        cohort_sizes = rep(3, 13),
```

```
                        chains = 2)
table(sims$recommended_dose) / length(sims$recommended_dose)
table(unlist(sims$doses_given)) / length(unlist(sims$doses_given))
table(unlist(sims$doses_given)) / length(sims$recommended_dose)

## End(Not run)
# In real life, we would run thousands of iterations, not 10.
# This is an example.
```

`efftox_solve_p` *Calculate the p-index for EffTox utility contours*

Description

Calculate the p-index for EffTox utility contours so that the neutral utility contour intersects the following points in the Prob(Efficacy) - Prob(Toxicity) plane: (`eff0`, 0), (1, `tox1`) and (`eff_star`, `tox_star`)

Usage

```
efftox_solve_p(eff0, tox1, eff_star, tox_star)
```

Arguments

`eff0`	Efficacy probability required when toxicity is impossible; a number between 0 and 1
`tox1`	Toxicity probability permitted when efficacy is guaranteed; a number between 0 and 1
`eff_star`	Efficacy probability of an equi-utility third point
`tox_star`	Toxicity probability of an equi-utility third point

Value

The p-index

References

Thall, Herrick, Nguyen, Venier & Norris. 2014, Effective sample size for computing prior hyper-parameters in Bayesian phase I-II dose-finding

Examples

```
efftox_solve_p(0.5, 0.65, 0.7, 0.25)
```

efftox_superiority *Get dose-superiority matrix in EffTox*

Description

Get a dose-superiority matrix from an EffTox dose analysis. EffTox seeks to choose the dose with the highest utility, thus superiority is inferred by posterior utility. The item in row i, col j is the posterior probability that the utility of dose j exceeds that of dose i.

Usage

```
efftox_superiority(fit)
```

Arguments

fit	An instance of efftox_fit.

Value

n by n matrix, where n is number of doses under investigation. The item in row i, col j is the posterior probability that the utility of dose j exceeds that of dose i.

Examples

```
fit <- stan_efftox_demo('1N 2E 3B')
sup_mat <- efftox_superiority(fit)
```

efftox_utility *Get the utility of efficacy & toxicity probability pairs*

Description

Get the utility of efficacy & toxicity probability pairs

Usage

```
efftox_utility(p, eff0, tox1, prob_eff, prob_tox)
```

Arguments

p	p-index of EffTox utility contours. Use efftox_solve_p
eff0	Efficacy probability required when toxicity is impossible; a number between 0 and 1
tox1	Toxicity probability permitted when efficacy is guaranteed; a number between 0 and 1
prob_eff	Probability of efficacy; number between 0 and 1
prob_tox	Probability of toxicity; number between 0 and 1

Value

Utility value(s)

See Also

efftox_solve_p

Examples

```
p <- efftox_solve_p(0.5, 0.65, 0.7, 0.25)

u <- efftox_utility(p, 0.5, 0.65, prob_eff = 0.7, prob_tox = 0.25)
round(u, 4) == 0

u <- efftox_utility(p, 0.5, 0.65, prob_eff = c(0.6, 0.7, 0.8),
                    prob_tox = c(0.1, 0.2, 0.3))
round(u, 2) == c(0.04, 0.08, 0.12)
```

---

efftox_utility_density_plot

Plot densities of EffTox dose utilities

Description

Plot densities of EffTox dose utilities. Optionally plot only a subset of the doses by specifying the doses parameter. This function requires ggplot2 be installed.

Usage

```
efftox_utility_density_plot(fit, doses = NULL)
```

Arguments

fit	An instance of efftox_fit.
doses	optional, vector of integer dose-levels to plot. E.g. to plot only dose-levels 1, 2 & 3 (and suppress the plotting of any other doses), use doses = 1:3

Value

an instance of ggplot. Omit assignment to just view the plot.

Note

This function requires that ggplot2 be installed.

Examples

```
fit <- stan_efftox_demo('1N 2E 3B')
efftox_utility_density_plot(fit) + ggplot2::ggtitle('My doses')  # Too busy?
# Specify subset of doses to make plot less cluttered
efftox_utility_density_plot(fit, doses = 1:3) + ggplot2::ggtitle('My doses')
```

`parse_dose_finding_outcomes` *Parse a string of dose-finding trial outcomes.*

Description

Parse a string of dose-finding trial outcomes

Parse a string of dose-finding trial outcomes to a list. The outcome string describes the doses given, outcomes observed and the timing of analyses that recommend a dose. The format of the string is the pure phase I analogue to that described in Brock _et al_. (2017). The letters T and N are used to represents patients that experienced (T)oxicity and (N)o toxicity. These letters are concatenated after numerical dose-levels to convey the outcomes of cohorts of patients. For instance, `2NNT` represents a cohort of three patients that were treated at dose-level 2, one of whom experienced toxicity, and two that did not. The results of cohorts are separated by spaces and it is assumed that a dose-finding decision takes place at the end of a cohort. Thus, `2NNT 1NN` builds on our previous example, where the next cohort of two were treated at dose-level 1 and neither of these patients experienced toxicity. See examples.

Usage

```
parse_dose_finding_outcomes(outcome_string)
```

Arguments

`outcome_string`	character representing doses given, outcomes observed, and timing of analyses. See Description.

Value

a list with a slot for each cohort. Each cohort slot is itself a list, containing elements: * `dose`, the integer dose delivered to the cohort; * `outcomes`, a character string representing the `T` or `N` outcomes for the patients in this cohort.

References

Brock, K., Billingham, L., Copland, M., Siddique, S., Sirovica, M., & Yap, C. (2017). Implementing the EffTox dose-finding design in the Matchpoint trial. BMC Medical Research Methodology, 17(1), 112. https://doi.org/10.1186/s12874-017-0381-x

Examples

```
x = parse_dose_finding_outcomes('1NNN 2NNT 3TT')
length(x)
x[[1]]$dose
x[[1]]$outcomes
x[[2]]$dose
x[[2]]$outcomes
x[[3]]$dose
x[[3]]$outcomes
```

---

peps2_get_data *Get data to run the PePS2 trial example*

Description

Get data to run the BEBOP model in the PePS2 trial. The trial investigates pembrolizumab in non-small-cell lung cancer. Patients may be previously treated (PT) or treatment naive (TN). Pembro response rates in lung cancer have been shown to increase with PD-L1 tumour proportion score. PD-L1 score is measured at baseline. Each patient belongs to one of the Low, Medium or High categories. These two baseline variables stratify the patient population and are used as predictive variables to stratify the analysis. The BEBOP model studies co-primary efficacy and toxicity outcomes in the presence of predictive data. Thus, PePS2 studies efficacy and toxicity in 6 distinct cohorts: TN Low, TN Medium, TN High, PT Low, PT Medium, PT High. The design admits all-comers and does not target specific sample sizes in the individual cohorts. Hyperprior parameters have defaults to match those used in PePS2, but all may be overridden. The returned object includes randomly-sampled outcomes, as well as parameters to run the model. These are all combined in the same list object for passing to RStan, as is the convention. See the accompanying vignette for a full description.

Usage

```
peps2_get_data(num_patients, cohort_probs = NULL, prob_eff, prob_tox,
  eff_tox_or, cohort_rho = c(15.7, 21.8, 12.4, 20.7, 18, 11.4),
  alpha_mean = -2.2, alpha_sd = 2, beta_mean = -0.5, beta_sd = 2,
  gamma_mean = -0.5, gamma_sd = 2, zeta_mean = -0.5, zeta_sd = 2,
  lambda_mean = -2.2, lambda_sd = 2, psi_mean = 0, psi_sd = 1)
```

Arguments

num_patients	Total number of patients to use, positive integer.
cohort_probs	Probabilities that a patient belongs to each of the 6 cohorts, in the order given above; a vector of numbers between 0 and 1 that add up to 1. cohort_probs or cohort_rho must be specified.
prob_eff	Probabilities of efficacy in each of the 6 cohorts, in the order given above; a vector of numbers between 0 and 1

prob_tox	Probabilities of toxicity in each of the 6 cohorts, in the order given above; a vector of numbers between 0 and 1
eff_tox_or	Measure of strength of association between efficacy and toxicity, in each of the 6 cohorts, in the order given above; a vector of numbers. Use 1 for no association; numbers increasingly greater than 1 for stronger positive associations, and numbers less than 1 for stronger negative associations
cohort_rho	Concentration parameters for cohort membership, in the order given above, using a Dirichlet distribution. This leads to randomly- sampled cohort sizes distributed Dir(cohort_rho). cohort_probs or cohort_rho must be specified.
alpha_mean	The prior mean of alpha. Alpha is the efficacy model intercept.
alpha_sd	The prior standard deviation of alpha. Alpha is the efficacy model intercept.
beta_mean	The prior mean of beta. Beta is the efficacy model term for being previously treated.
beta_sd	The prior standard deviation of beta. Beta is the efficacy model term for being previously treated.
gamma_mean	The prior mean of gamma. Gamma is the efficacy model term for being PD-L1 score = Low.
gamma_sd	The prior standard deviation of gamma. Gamma is the efficacy model term for being PD-L1 score = Low.
zeta_mean	The prior mean of zeta. Zeta is the efficacy model term for being PD-L1 score = Medium.
zeta_sd	The prior standard deviation of zeta. Zeta is the efficacy model term for being PD-L1 score = Medium.
lambda_mean	The prior mean of lambda. Lambda is the toxicity model intercept.
lambda_sd	The prior standard deviation of lambda. Lambda is the toxicity model intercept.
psi_mean	The prior mean of psi. Psi is the joint model association parameter.
psi_sd	The prior standard deviation of psi. Psi is the joint model association parameter.

Value

a list of parameters

Examples

```
set.seed(123)
dat <- peps2_get_data(num_patients = 60,
                      prob_eff = c(0.167, 0.192, 0.5, 0.091, 0.156, 0.439),
                      prob_tox = rep(0.1, 6),
                      eff_tox_or = rep(1, 6))
fit <- stan_peps2(
  eff = dat$eff,
  tox = dat$tox,
  cohorts = dat$cohorts
)
```

peps2_process *Process RStan samples from a BEBOP model fit to PePS2 data*

Description

Process RStan samples from a BEBOP model fit to PePS2 data. This step lets us make inferences about whether the modelled efficacy and toxicity probabilities suggest the treatment is acceptable in each of the cohorts under study. The parameters have default values to match those used in the PePS2 trial. See the accompanying vignette for a full description.

Usage

```
peps2_process(fit, min_eff = 0.1, max_tox = 0.3, eff_cert = 0.7,
  tox_cert = 0.9)
```

Arguments

fit	An instance of rstan::stanmodel, derived by fitting data to the BEBOP in PePS2 model. Use stan_peps2.
min_eff	The lower efficacy probability threshold; a number between 0 and 1.
max_tox	The upper toxicity probability threshold; a number between 0 and 1.
eff_cert	Certainty required to infer the treatment is acceptable with regards to being probably efficacious; a number between 0 and 1.
tox_cert	Certainty required to infer the treatment is acceptable with regards to being probably tolerable; a number between 0 and 1.

Value

a list with the following items:

- ProbEff, the posterior mean probability of efficacy in the 6 cohorts.
- ProbAccEff, the posterior mean probability that the probability of efficacy exceeds min_eff, in the 6 cohorts.
- ProbTox, the posterior mean probability of toxicity in the 6 cohorts.
- ProbAccTox, the posterior mean probability that the probability of toxicity is less than max_tox, in the 6 cohorts.
- Accept, a vector of logical values to show whether treatment should be accepted in the 6 cohorts. Treatment is acceptable when it is probably efficacious and probably not toxic, with respect to the described rules.
- alpha, the posterior mean estimate of alpha.
- beta, the posterior mean estimate of beta.
- gamma, the posterior mean estimate of gamma.
- zeta, the posterior mean estimate of zeta.
- lambda, the posterior mean estimate of lambda.
- psi, the posterior mean estimate of psi.

See Also

peps2_get_data

Examples

```
set.seed(123)
fit <- stan_peps2(
  eff = c(0, 1, 0, 1, 0, 0),
  tox = c(0, 0, 1, 1, 0, 0),
  cohorts = c(3, 1, 1, 4, 5, 6)
)
decision <- peps2_process(fit)
decision$Accept
decision$ProbEff
decision$ProbAccEff
```

---

plot.crm_fit *Plot an crm_fit*

Description

Plot an crm_fit

Usage

```
## S3 method for class 'crm_fit'
plot(x, pars = "prob_tox", ...)
```

Arguments

x	crm_fit object to plot.
pars	Parameters to plot. Plots utility scores by default.
...	Extra parameters, passed onwards.

Value

A plot

plot.efftox_fit *Plot an efftox_fit*

Description

Plot an efftox_fit

Usage

```
## S3 method for class 'efftox_fit'
plot(x, pars = "utility", ...)
```

Arguments

x	efftox_fit object to plot.
pars	Parameters to plot. Plots utility scores by default.
...	Extra parameters, passed onwards.

Value

A plot

print.crm_fit *Print crm_fit object.*

Description

Print crm_fit object.

Usage

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

Arguments

x	crm_fit object to convert.
...	Extra parameters, passed onwards.

print.efftox_fit *Print efftox_fit object.*

Description

Print efftox_fit object.

Usage

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

Arguments

x	efftox_fit object to convert.
...	Extra parameters, passed onwards.

ranBin2 *Sample pairs of correlated binary events*

Description

This function is reproduced from the binarySimCLF package on CRAN. The original package appears no longer to be maintained. View the original source at: https://github.com/cran/binarySimCLF/blob/master/R/ranBin2.

Usage

```
ranBin2(nRep, u, psi)
```

Arguments

nRep	Number of simulated event pairs, positive integer.
u	Mean event probabilities, expressed as a vector of length 2. E.g. to simulate associated bivariate events with probabilities 80 `u = c(0.8, 0.3)`.
psi	Odds ratio, number. This parameter controls the strength of association. Use `psi = 1` for no association. Values greater than 1 correspond to increasingly positive association between the two events, and vice-versa.

Value

Matrix of events represented as 0s and 1s, with nRep rows and 2 columns. The first column is the incidence of event 1.

Examples

```
probs <- c(0.8, 0.3)
s <- ranBin2(1000, probs, psi=0.2)  # 1000 pairs of outcomes
cor(s)  # Negatively correlated because psi < 1
colMeans(s)  # Event rates as expected
```

`stan_crm` *Fit a CRM model*

Description

Fit a CRM model using Stan for full Bayesian inference.

Usage

```
stan_crm(outcome_str = NULL, skeleton, target, model = c("empiric",
  "logistic", "logistic_gamma", "logistic2"), a0 = NULL, alpha_mean = NULL,
  alpha_sd = NULL, beta_mean = NULL, beta_sd = NULL, beta_shape = NULL,
  beta_inverse_scale = NULL, doses_given = NULL, tox = NULL,
  weights = NULL, ...)
```

Arguments

`outcome_str`	A string representing the outcomes observed hitherto. See `df_parse_outcomes` for a description of syntax and examples. Alternatively, you may provide `doses_given` and `tox` parameters. See Details.
`skeleton`	a vector of the prior guesses of toxicity at doses. This should be a monotonically-increasing vector of numbers between 0 and 1.
`target`	the target toxicity probability, a number between 0 and 1. This value would normally be one of the values in `skeleton`, but that is not a requirement.
`model`	Character string to denote desired model. One of `empiric`, `logistic`, `logistic_gamma`, or `logistic2`. The choice of model determines which parameters are required. See Details.
`a0`	Value of fixed intercept parameter. Only required for certain models. See Details.
`alpha_mean`	Prior mean of intercept variable for normal prior. Only required for certain models. See Details.
`alpha_sd`	Prior standard deviation of intercept variable for normal prior. Only required for certain models. See Details.
`beta_mean`	Prior mean of gradient variable for normal prior. Only required for certain models. See Details.
`beta_sd`	Prior standard deviation of slope variable for normal prior. Only required for certain models. See Details.
`beta_shape`	Prior shape parameter of slope variable for gamma prior. Only required for certain models. See Details.

`beta_inverse_scale`
: Prior inverse scale parameter of slope variable for gamma prior. Only required for certain models. See Details.

`doses_given`
: A optional vector of dose-levels given to patients 1:num_patients, where 1=lowest dose, 2=second dose, etc. Only required when `outcome_str` is not provided.

`tox`
: An optional vector of toxicity outcomes for patients 1:num_patients, where 1=toxicity and 0=no toxicity. Only required when `outcome_str` is not provided.

`weights`
: An optional vector of numeric weights for the observations for patients 1:num_patients, thus facilitating the TITE-CRM design. Can be used with `outcome_str`, or with `doses_given` and `tox`. It is generally tider to specify `doses_given`, `tox` and `weights` when a TITE-CRM analysis is desired.

`...`
: Extra parameters are passed to `rstan::sampling`. Commonly used options are `iter`, `chains`, `warmup`, `cores`, and `control`.

Details

The quickest and easiest way to fit a CRM model to some observed outcomes is to describe the outcomes using **trialr**'s syntax for dose-finding outcomes. See `df_parse_outcomes` for full details and examples.

Different model parameterisations require that difference parameter values are specified.

Value

An object of class `crm_fit`

Requirements of `empiric` model

* beta_sd

Requirements of `logistic` model

* a0 * beta_mean * beta_sd

Requirements of `logistic_gamma` model

* a0 * beta_shape * beta_inverse_scale

Requirements of `logistic2` model

* a0 * alpha_mean * alpha_sd * beta_mean * beta_sd

Author(s)

Kristian Brock <kristian.brock@gmail.com>

References

O'Quigley, J., Pepe, M., & Fisher, L. (1990). Continual reassessment method: a practical design for phase 1 clinical trials in cancer. Biometrics, 46(1), 33-48. https://doi.org/10.2307/2531628

See Also

crm_fit sampling.

Examples

```
## Not run:
# CRM example
mod1 <- stan_crm('1N 2N 3T', skeleton = c(0.1, 0.2, 0.35, 0.6),
                 target = 0.2, model = 'empiric', beta_sd = sqrt(1.34),
                 seed = 123)

mod2 <- stan_crm('1NNN 2NNN 3TTT', skeleton = c(0.1, 0.2, 0.35, 0.6),
                 target = 0.2, model = 'logistic', a0 = 3, beta_mean = 0,
                 beta_sd = sqrt(1.34), seed = 123)

# The seed is passed to the Stan sampler. The usual Stan sampler params like
# cores, iter, chains etc are passed on too via the ellipsis operator.

# TITE-CRM example, p.124 of Dose Finding by the CRM, Cheung (2010)
mod3 <-stan_crm(skeleton = c(0.05, 0.12, 0.25, 0.40, 0.55), target = 0.25,
                doses_given = c(3, 3, 3, 3),
                tox = c(0, 0, 0, 0),
                weights = c(73, 66, 35, 28) / 126,
                model = 'empiric', beta_sd = sqrt(1.34), seed = 123)
mod3$recommended_dose

## End(Not run)
```

---

stan_efftox *Fit an EffTox model*

Description

Fit an EffTox model using Stan for full Bayesian inference.

Usage

```
stan_efftox(outcome_str = NULL, real_doses, efficacy_hurdle, toxicity_hurdle,
  p_e, p_t, eff0, tox1, eff_star, tox_star, alpha_mean, alpha_sd, beta_mean,
  beta_sd, gamma_mean, gamma_sd, zeta_mean, zeta_sd, eta_mean, eta_sd, psi_mean,
  psi_sd, doses_given = NULL, eff = NULL, tox = NULL, ...)
```

Arguments

outcome_str	A string representing the outcomes observed hitherto. See efftox_parse_outcomes for a description of syntax and examples. Alternatively, you may provide doses_given, eff and tox parameters. See Details.

`real_doses` A vector of numbers.The doses under investigation. They should be ordered from lowest to highest and be in consistent units. E.g., #' to conduct a dose-finding trial of doses 10mg, 20mg and 50mg, use c(10, 20, 50).

`efficacy_hurdle` Minimum acceptable efficacy probability. A number between 0 and 1.

`toxicity_hurdle` Maximum acceptable toxicity probability. A number between 0 and 1.

`p_e` Certainty required to infer a dose is acceptable with regards to being probably efficacious; a number between 0 and 1.

`p_t` Certainty required to infer a dose is acceptable with regards to being probably tolerable; a number between 0 and 1.

`eff0` Efficacy probability required when toxicity is impossible; a number between 0 and 1 (see Details).

`tox1` Toxicity probability permitted when efficacy is guaranteed; a number between 0 and 1 (see Details).

`eff_star` Efficacy probability of an equi-utility third point (see Details).

`tox_star` Toxicity probability of an equi-utility third point (see Details).

`alpha_mean` The prior normal mean of the intercept term in the toxicity logit model. A number.

`alpha_sd` The prior normal standard deviation of the intercept term in the toxicity logit model. A number.

`beta_mean` The prior normal mean of the slope term in the toxicity logit model. A number.

`beta_sd` The prior normal standard deviation of the slope term in the toxicity logit model. A number.

`gamma_mean` The prior normal mean of the intercept term in the efficacy logit model. A number.

`gamma_sd` The prior normal standard deviation of the intercept term in the efficacy logit model. A number.

`zeta_mean` The prior normal mean of the slope term in the efficacy logit model. A number.

`zeta_sd` The prior normal standard deviation of the slope term in the efficacy logit model. A number.

`eta_mean` The prior normal mean of the squared term coefficient in the efficacy logit model. A number.

`eta_sd` The prior normal standard deviation of the squared term coefficient in the efficacy logit model. A number.

`psi_mean` The prior normal mean of the association term in the combined efficacy-toxicity model. A number.

`psi_sd` The prior normal standard deviation of the association term in the combined efficacy-toxicity model. A number.

`doses_given` A optional vector of dose-levels given to patients 1:num_patients, where 1=lowest dose, 2=second dose, etc. Only required when `outcome_str` is not provided.

`eff`	An optional vector of efficacy outcomes for patients 1:num_patients, where 1=efficacy and 0=no efficacy. Only required when `outcome_str` is not provided.
`tox`	An optional vector of toxicity outcomes for patients 1:num_patients, where 1=toxicity and 0=no toxicity. Only required when `outcome_str` is not provided.
`...`	Extra parameters are passed to `rstan::sampling`. Commonly used options are `iter`, `chains`, `warmup`, `cores`, `control`. `sampling`.

Details

The quickest and easiest way to fit an EffTox model to some observed outcomes is to describe the outcomes using **trialr**'s syntax for efficacy-toxicity dose-finding outcomes. See `efftox_parse_outcomes` for full details and examples.

Utility or attractivess scores are calculated in EffTox using L^p norms. Imagine the first quadrant of a scatter plot with prob_eff along the x-axis and prob_tox along the y-axis. The point (1, 0) (i.e. guaranteed efficacy & no toxicity) is the holy grail. The neutral contour intersects the points (eff0, 0), (1, tox1) and (eff_star, tox_star). A unique curve intersects these three points and identifies a value for p, the exponent in the L^p norm. On this neutral- utility contour, scores are equal to zero. A family of curves with different utility scores is defined that are "parallel" to this neutral curve. Points with probabilities of efficacy and toxicity that are nearer to (1, 0) will yield greater scores, and vice-versa.

Value

An object of class `efftox_fit`

Author(s)

Kristian Brock <kristian.brock@gmail.com>

References

Thall, P., & Cook, J. (2004). Dose-Finding Based on Efficacy-Toxicity Trade-Offs. Biometrics, 60(3), 684-693.

Thall, P., Herrick, R., Nguyen, H., Venier, J., & Norris, J. (2014). Effective sample size for computing prior hyperparameters in Bayesian phase I-II dose-finding. Clinical Trials, 11(6), 657-666. https://doi.org/10.1177/1740774514547397

Brock, K., Billingham, L., Copland, M., Siddique, S., Sirovica, M., & Yap, C. (2017). Implementing the EffTox dose-finding design in the Matchpoint trial. BMC Medical Research Methodology, 17(1), 112. https://doi.org/10.1186/s12874-017-0381-x

See Also

`efftox_fit` `stan_efftox_demo`

Examples

```
## Not run:
# This model is presented in Thall et al. (2014)
mod1 <- stan_efftox('1N 2E 3B',
                    real_doses = c(1.0, 2.0, 4.0, 6.6, 10.0),
                    efficacy_hurdle = 0.5, toxicity_hurdle = 0.3,
                    p_e = 0.1, p_t = 0.1,
                    eff0 = 0.5, tox1 = 0.65,
                    eff_star = 0.7, tox_star = 0.25,
                    alpha_mean = -7.9593, alpha_sd = 3.5487,
                    beta_mean = 1.5482, beta_sd = 3.5018,
                    gamma_mean = 0.7367, gamma_sd = 2.5423,
                    zeta_mean = 3.4181, zeta_sd = 2.4406,
                    eta_mean = 0, eta_sd = 0.2,
                    psi_mean = 0, psi_sd = 1, seed = 123)

# Shorthand for the above is:
mod2 <- stan_efftox_demo('1N 2E 3B', seed = 123)

# the seed is passed to the Stan sampler. The usual Stan sampler params like
# cores, iter, chains etc are passed on too via the ellipsis operator.

## End(Not run)
```

`stan_efftox_demo` *Fit the EffTox model presented in Thall et al. (2014)*

Description

Fit the EffTox model presented in Thall et al. (2014) using Stan for full Bayesian inference.

Usage

```
stan_efftox_demo(outcome_str, ...)
```

Arguments

`outcome_str`	A string representing the outcomes observed hitherto. See `efftox_parse_outcomes` for a description of syntax and examples. Alternatively, you may provide `doses_given`, `eff` and `tox` parameters. See Details.
`...`	Extra parameters are passed to `rstan::sampling`. Commonly used options are `iter`, `chains`, `warmup`, `cores`, `control`. `sampling`.

Value

An object of class `efftox_fit`

Author(s)

Kristian Brock <kristian.brock@gmail.com>

References

Thall, P., & Cook, J. (2004). Dose-Finding Based on Efficacy-Toxicity Trade-Offs. Biometrics, 60(3), 684-693.

Thall, P., Herrick, R., Nguyen, H., Venier, J., & Norris, J. (2014). Effective sample size for computing prior hyperparameters in Bayesian phase I-II dose-finding. Clinical Trials, 11(6), 657-666. https://doi.org/10.1177/1740774514547397

Brock, K., Billingham, L., Copland, M., Siddique, S., Sirovica, M., & Yap, C. (2017). Implementing the EffTox dose-finding design in the Matchpoint trial. BMC Medical Research Methodology, 17(1), 112. https://doi.org/10.1186/s12874-017-0381-x

See Also

efftox_fit stan_efftox

Examples

```
## Not run:
# This model is presented in Thall et al. (2014)
mod2 <- stan_efftox_demo('1N 2E 3B', seed = 123)

# The seed is passed to the Stan sampler. The usual Stan sampler params like
# cores, iter, chains etc are passed on too via the ellipsis operator.

## End(Not run)
```

---

stan_hierarchical_response_thall

Fit the hierarchical response model described by Thall et al. *(2003).*

Description

Fit the hierarchical response model to exchangeable groups described by Thall *et al.* (2003).

Usage

```
stan_hierarchical_response_thall(group_responses, group_sizes, mu_mean, mu_sd,
  tau_alpha, tau_beta, ...)
```

Arguments

group_responses	vector of integers, number of responses in each group
group_sizes	vector of integers, number of patients in each group
mu_mean	mean parameter of normal prior distribution on mu. See details.
mu_sd	standard deviation parameter of normal prior distribution on mu. See details.
tau_alpha	parameter alpha of inverse gamma prior distribution on tau. See details.
tau_beta	beta parameter of inverse gamma prior distribution on tau. See details.
...	Extra parameters are passed to `rstan::sampling`. Commonly used options are `iter`, `chains`, `warmup`, `cores`, and `control`.

Details

Thall *et al.* (2003) describe hierarchical methods for analysing treatment effects of a common intervention in several sub-types of a disease. The treatment effects are assumed to be different but exchangeable and correlated. Observing efficacy in one cohort, for example, increases one's expectations of efficacy in others. They demonstrate the hierarchical approach in a trial with binary response outcomes and in another with time-to-event outcomes. This function fits their model for binary response outcomes.

Let the probability of response in group i be $\pi[i]$ for $i = 1, ..., N$. They assume a logistic model so that $\theta[i] = log\pi[i]/(1 - \pi[i])$ is the log-odds of response in group i. They assume that $\theta[i]$ $N(\mu, \sigma^2)$.

The authors implemented their model in BUGS. As is the convention in BUGS, the authors define normal distributions by a precision parameter τ as opposed to the standard deviation parameter σ used here. We have re-specified their model to comply with the Stan convention of using standard deviation. The authors use a normal prior on μ, and a gamma prior on τ, equivalent to an inverse gamma prior on $\tau^{-1} = \sigma^2$.

The authors provide WinBUGS code in their publication. We implement their model here in Stan.

Value

Object of class `rstan::stanfit` returned by `rstan::sampling`

References

Thall, Wathen, Bekele, Champlin, Baker, and Benjamin. 2003. "Hierarchical Bayesian approaches to phase II trials in diseases with multiple subtypes." Statistics in Medicine 22 (5): 763–80. https://doi.org/10.1002/sim.1399.

See Also

`rstan::stanfit`, `rstan::sampling`

Examples

```
## Not run:
# Example from p.778 of Thall et al. (2003)
mod0 <- stan_hierarchical_response_thall(
  group_responses = c(0, 0, 1, 3, 5, 0, 1, 2, 0, 0),
```

```
  group_sizes = c(0, 2 ,1, 7, 5, 0, 2, 3, 1, 0),
  mu_mean = -1.3863,
  mu_sd = sqrt(1 / 0.1),
  tau_alpha = 2,
  tau_beta = 20)

## End(Not run)
```

---

stan_peps2 *Fit the P2TNE model developed for the PePS2 trial to some outcomes.*

Description

The PePS2 trial investigates pembrolizumab in non-small-cell lung cancer. Patients may be previously treated (PT) or treatment naive (TN). Response rates in lung cancer have been shown to increase with PD-L1 tumour proportion score. PD-L1 score is measured at baseline. Each patient belongs to one of the categories <1 stratify the patient population and are used as predictive variables to stratify the analysis. The BEBOP model studies co-primary efficacy and toxicity outcomes in the presence of predictive data. Thus, PePS2 studies efficacy and toxicity in 6 distinct cohorts: TN Low, TN Medium, TN High, PT Low, PT Medium, PT High. The design admits all-comers and does not target specific sample sizes in the individual cohorts. Hyperprior parameters have defaults to match those used in PePS2, but all may be overridden. The returned object includes randomly-sampled outcomes, as well as parameters to run the model. These are all combined in the same list object for passing to RStan, as is the convention. See the accompanying vignette for a full description.

Usage

```
stan_peps2(eff, tox, cohorts, alpha_mean = -2.2, alpha_sd = 2,
  beta_mean = -0.5, beta_sd = 2, gamma_mean = -0.5, gamma_sd = 2,
  zeta_mean = -0.5, zeta_sd = 2, lambda_mean = -2.2, lambda_sd = 2,
  psi_mean = 0, psi_sd = 1, ...)
```

Arguments

eff	A vector of efficacy outcomes for the patients, where 1=efficacy and 0=no efficacy.
tox	A vector of toxicity outcomes for the patients, where 1=toxicity and 0=no toxicity.
cohorts	A vector of integers from 1 to 6, denoting the cohorts to which the patients belong.
alpha_mean	The prior mean of alpha. Alpha is the efficacy model intercept.
alpha_sd	The prior standard deviation of alpha. Alpha is the efficacy model intercept.
beta_mean	The prior mean of beta. Beta is the efficacy model term for being previously treated.

beta_sd	The prior standard deviation of beta. Beta is the efficacy model term for being previously treated.
gamma_mean	The prior mean of gamma. Gamma is the efficacy model term for being PD-L1 score = Low.
gamma_sd	The prior standard deviation of gamma. Gamma is the efficacy model term for being PD-L1 score = Low.
zeta_mean	The prior mean of zeta. Zeta is the efficacy model term for being PD-L1 score = Medium.
zeta_sd	The prior standard deviation of zeta. Zeta is the efficacy model term for being PD-L1 score = Medium.
lambda_mean	The prior mean of lambda. Lambda is the toxicity model intercept.
lambda_sd	The prior standard deviation of lambda. Lambda is the toxicity model intercept.
psi_mean	The prior mean of psi. Psi is the joint model association parameter.
psi_sd	The prior standard deviation of psi. Psi is the joint model association parameter.
...	Extra parameters are passed to `rstan::sampling`. Commonly used options are `iter`, `chains`, `warmup`, `cores`, and `control`.

Value

Object of class `rstan::stanfit` returned by `rstan::sampling`

Examples

```
## Not run:
fit <- stan_peps2(
  eff = c(0, 1, 0, 1, 0, 0),
  tox = c(0, 0, 1, 1, 0, 0),
  cohorts = c(3, 1, 1, 4, 5, 6)
)

## End(Not run)
```

---

summary.crm_fit *Obtain summary of an crm_fit*

Description

Obtain summary of an crm_fit

Usage

```
## S3 method for class 'crm_fit'
summary(object, ...)
```

Arguments

object	crm_fit object to summarise.
...	Extra parameters, passed onwards.

Value

A summary object.

---

summary.efftox_fit *Obtain summary of an efftox_fit*

Description

Obtain summary of an efftox_fit

Usage

```
## S3 method for class 'efftox_fit'
summary(object, ...)
```

Arguments

object	efftox_fit object to summarise.
...	Extra parameters, passed onwards.

Value

A summary object.

Index