Package: snSMART (via r-universe)

October 17, 2024

Type Package

Title Small N Sequential Multiple Assignment Randomized Trial Methods

Version 0.2.4

Maintainer Michael Kleinsasser <mkleinsa@umich.edu>

Description Consolidated data simulation, sample size calculation and analysis functions for several snSMART (small sample sequential, multiple assignment, randomized trial) designs under one library. See Wei, B., Braun, T.M., Tamura, R.N. and Kidwell, K.M. ``A Bayesian analysis of small n sequential multiple assignment randomized trials (snSMARTs)." (2018) Statistics in medicine, 37(26), pp.3723-3732 <doi:10.1002/sim.7900>.

License GPL (>= 2)

URL https://github.com/sidiwang/snSMART

BugReports https://github.com/sidiwang/snSMART/issues

Depends R (>= 3.5.0), EnvStats (>= 2.4.0)

Imports bayestestR (>= 0.11.0), condMVNorm (>= 2020.1), cubature (>= 2.0.4.1), geepack (>= 1.3-1), HDInterval (>= 0.2.0), pracma (>= 2.3.3), rjags (>= 4-12), tidyr (>= 1.1.2), truncdist (>= 1.0-1)

Suggests coda (>= 0.19-2), testthat (>= 3.0.0)

Biarch true

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

SystemRequirements JAGS 4.x.y

Config/testthat/edition 3

NeedsCompilation no

Author Sidi Wang [aut], Kelley Kidwell [aut], Michael Kleinsasser [cre]

Repository CRAN

Date/Publication 2024-10-16 20:00:02 UTC

Contents

`BJSM_binary` *BJSM for snSMART (3 active treatments/placebo and 2 dose level) with binary outcome*

Description

This function implements the BJSM (Bayesian Joint Stage Modeling) method which borrows information across both stages to estimate the individual response rate of each treatment/dose level in a snSMART design with binary outcomes.

Usage

```
BJSM_binary(
  data,
  prior_dist,
  pi_prior,
  normal.par,
  beta_prior,
  n_MCMC_chain = 1,
  n.adapt,
  BURN.IN = 100,
  thin = 1,
  MCMC_SAMPLE,
  ci = 0.95,
  six = TRUE,
  DTR = TRUE,
  jags.model_options = NULL,
  coda.samples_options = NULL,
  verbose = FALSE,
  ...
```

```
)

## S3 method for class 'summary.BJSM_binary'
print(x, ...)

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

## S3 method for class 'summary.BJSM_dose_binary'
print(x, ...)

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

Arguments

data	trial data with 4 columns: treatment_stageI, response_stageI,treatment_stageII and response_stageII. Missing data is allowed in stage 2.
prior_dist	for 3 active treatment design: vector of three values ("prior distribution for pi", "prior distribution for beta0", "prior distribution for beta1"). User can choose from "gamma", "beta", "pareto". e.g. prior_dist = c("beta", "beta", "pareto"); for dose level design: vector of two values ("prior distribution for pi_P", "prior distribution for beta")
pi_prior	for 3 active treatment design: vector of six values (a, b, c, d, e, f), where a and b are the parameter a and parameter b of the prior distribution for pi_1A, c and d are the parameter a and parameter b of the prior distribution for pi_1B, and e and f are the parameter a and parameter b of the prior distribution for pi_1C. for dose level design: vector of two values (a, b). a is the parameter a of the prior distribution for pi (response rate) of placebo. b is the parameter b of the prior distribution for pi of placebo. Please check the Details section for more explanation
normal.par	for dose level design: vector of two values (normal.mean, normal.var). our function assumes that the logarithm of treatment effect ratio follows a Gaussian prior distribution $N(\mu, \sigma^2)$, that is $log(\pi_L/\pi_P)\ N(normal.mean, normal.var)$, and $log(\pi_H/\pi_P)\ N(normal.mean, normal.var)$. normal.mean is the mean of this Gaussian prior. normal.var is the variance of this Gaussian prior distribution
beta_prior	for 3 active treatment design: vector of four values (a, b, c, d). a is the value of parameter a of the prior distribution for linkage parameter beta_0 or beta_0m, b is the value of parameter b of the prior distribution for linkage parameter beta_0 or beta_0m. c is the value of parameter a of the prior distribution for linkage parameter beta_1 or beta_1m. d is the value of parameter b of the prior distribution for linkage parameter beta_1 or beta_1m. for dose level design: vector of two values (a, b). a is the parameter a of the prior distribution for linkage parameter beta. b is the parameter b of the prior distribution for linkage parameter beta. Please check the Details section for more explanation
n_MCMC_chain	number of MCMC chains, default to 1.

n.adapt	the number of iterations for adaptation
BURN.IN	number of burn-in iterations for MCMC
thin	thinning interval for monitors
MCMC_SAMPLE	number of iterations for MCMC
ci	coverage probability for credible intervals, default = 0.95
six	TRUE or FALSE. If TRUE, will run the six beta model (allow for estimating beta_0m and beta_1m values that differ among different treatments m), if FALSE will run the two beta model. default = TRUE. Only need to specify this for 3 active treatment design.
DTR	TRUE or FALSE. If TRUE, will also return the expected response rate of dynamic treatment regimens. default = TRUE. Only need to specify this for 3 active treatment design.
jags.model_options	a list of optional arguments that are passed to jags.model() function.
coda.samples_options	a list of optional arguments that are passed to coda.samples() function.
verbose	TRUE or FALSE. If FALSE, no function message and progress bar will be printed.
...	further arguments. Not currently used.
x	object to summarize.

Details

For gamma distribution, prior.a is the shape parameter r, prior.b is the rate parameter lambda. For beta distribution, prior.a is the shape parameter a, prior.b is the shape parameter b. For pareto distribution, prior.a is the scale parameter alpha, prior.b is the shape parameter c (see jags user manual).

The individual response rate is regarded as a permanent feature of the treatment. The second stage outcome is modeled conditionally on the first stage results linking the first and second stage response probabilities through linkage parameters. The first stage response rate is denoted as π_m for treatment m. In the two β model, the second stage response rate for first stage responders is equal to $\beta_1\pi_m$. For nonresponders to treatment m in the first stage who receive treatment m' in the second the stage, the second stage response rate in the second stage is equal to $\beta_0\pi_{m'}$. In the six β model, the second stage response rate of the first stage responders to treatment m is denoted by $\beta_{1m}\pi_m$, and the second stage response rate of the non-responders to first stage treatment m who receive treatment m' in the second stage is denoted by $\beta_{0m}\pi_{m'}$. All the βs are linkage parameters.

Please refer to the paper listed under reference section for standard snSMART trial design and detailed definition of parameters.

Note that this package does not include the JAGS library, users need to install JAGS separately. Please check this page for more details: https://sourceforge.net/projects/mcmc-jags/

Value

posterior_sample

an mcmc.list object generated through the coda.samples() function, which includes posterior samples of the link parameters and response rates generated through the MCMC process

`pi_hat_bjsm` estimate of response rate/treatment effect

`se_hat_bjsm` standard error of the response rate

`ci_pi_A(P)`, `ci_pi_B(L)`, `ci_pi_C(H)`
x% credible intervals for treatment A(P), B(L), C(H)

`diff_AB(PL)`, `diff_BC(LH)`. `diff_AC(PH)`
estimate of differences between treatments A(P) and B(L), B(L) and C(H), A(P) and C(H)

`ci_diff_AB(PL)`, `ci_diff_BC(LH)`, `ci_diff_AC(PH)`
x% credible intervals for the estimated differences between treatments A(P) and B(L), B(L) and C(H), A(P) and C(H)

`se_AB(PL)`, `se_BC(LH)`, `se_AC(PH)`
standard error for the estimated differences between treatments A(P) and B(L), B(L) and C(H), A(P) and C(H)

`beta0_hat`, `beta1_hat`
linkage parameter `beta0` and `beta1` estimates

`se_beta0_hat`, `se_beta1_hat`
standard error of the estimated value of linkage parameter `beta0` and `beta1`

`ci_beta0_hat`, `ci_beta1_hat`
linkage parameter `beta0` and `beta1` credible interval

`pi_DTR_est` expected response rate of dynamic treatment regimens (DTRs)

`pi_DTR_se` standard error for the estimated DTR response rate

`ci_pi_AB`, `ci_pi_AC`, `ci_pi_BA`, `ci_pi_BC`, `ci_pi_CA`, `ci_pi_CB`
x% credible intervals for the estimated DTR response rate

References

Wei, B., Braun, T.M., Tamura, R.N. and Kidwell, K.M., 2018. A Bayesian analysis of small n sequential multiple assignment randomized trials (snSMARTs). Statistics in medicine, 37(26), pp.3723-3732. URL: doi:10.1002/sim.7900

Chao, Y.C., Trachtman, H., Gipson, D.S., Spino, C., Braun, T.M. and Kidwell, K.M., 2020. Dynamic treatment regimens in small n, sequential, multiple assignment, randomized trials: An application in focal segmental glomerulosclerosis. Contemporary clinical trials, 92, p.105989. URL: doi:10.1016/j.cct.2020.105989

Fang, F., Hochstedler, K.A., Tamura, R.N., Braun, T.M. and Kidwell, K.M., 2021. Bayesian methods to compare dose levels with placebo in a small n, sequential, multiple assignment, randomized trial. Statistics in Medicine, 40(4), pp.963-977. URL: doi:10.1002/sim.8813

See Also

`LPJSM_binary`
`sample_size`

Examples

```
mydata <- data_binary

BJSM_result <- BJSM_binary(
  data = mydata, prior_dist = c("beta", "beta", "pareto"),
  pi_prior = c(0.4, 1.6, 0.4, 1.6, 0.4, 1.6), beta_prior = c(1.6, 0.4, 3, 1),
  n_MCMC_chain = 1, n.adapt = 1000, MCMC_SAMPLE = 2000, ci = 0.95,
  six = TRUE, DTR = TRUE, verbose = FALSE
)


BJSM_result2 <- BJSM_binary(
  data = mydata, prior_dist = c("beta", "beta", "pareto"),
  pi_prior = c(0.4, 1.6, 0.4, 1.6, 0.4, 1.6), beta_prior = c(1.6, 0.4, 3, 1),
  n_MCMC_chain = 1, n.adapt = 10000, MCMC_SAMPLE = 60000, ci = 0.95,
  six = FALSE, DTR = FALSE, verbose = FALSE
)

summary(BJSM_result)
summary(BJSM_result2)


data <- data_dose
BJSM_dose_result <- BJSM_binary(
  data = data_dose, prior_dist = c("beta", "gamma"),
  pi_prior = c(3, 17), normal.par = c(0.2, 100), beta_prior = c(2, 2),
  n_MCMC_chain = 2, n.adapt = 1000, MCMC_SAMPLE = 6000, ci = 0.95, verbose = FALSE
)

summary(BJSM_dose_result)
```

BJSM_c *BJSM continuous (snSMART with three active treatments and a continuous outcome design)*

Description

BJSM (Bayesian Joint Stage Modeling) method that borrows information across both stages to estimate the individual response rate of each treatment (with continuous outcome and a mapping function).

Usage

```
BJSM_c(
  data,
  xi_prior.mean,
  xi_prior.sd,
  phi3_prior.sd,
```

```
  n_MCMC_chain,
  n.adapt,
  MCMC_SAMPLE,
  ci = 0.95,
  n.digits,
  thin = 1,
  BURN.IN = 100,
  jags.model_options = NULL,
  coda.samples_options = NULL,
  verbose = FALSE,
  ...
)

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

## S3 method for class 'summary.BJSM_c'
print(x, ...)

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

Arguments

`data`	trial ddatset with columns: `id`, `trt1` (treatment 1), `stage1outcome`, `stay` (stay = 1 if patient stay on the same treatment in stage 2, otherwise stay = 0), `trt2` (treatment 2), `stage2outcome`
`xi_prior.mean`	a 3-element vector of mean of the prior distributions (normal distribution) for `xi`s (treatment effect). Please check the `Details` section for more explaination
`xi_prior.sd`	a 3-element vector of standard deviation of the prior distributions (normal distribution) for `xi`s (treatment effect). Please check the `Details` section for more explaination
`phi3_prior.sd`	standard deviation of the prior distribution (folded normal distribution) of `phi3` (if the patient stays on the same treatment, `phi3` is the cumulative effect of stage 1 that occurs on the treatment longer term). Please check the `Details` section for more explaination
`n_MCMC_chain`	number of MCMC chains, default to 1
`n.adapt`	the number of iterations for adaptation
`MCMC_SAMPLE`	number of iterations for MCMC
`ci`	coverage probability for credible intervals, default = 0.95
`n.digits`	number of digits to keep in the final estimation of treatment effect
`thin`	thinning interval for monitors
`BURN.IN`	number of burn-in iterations for MCMC
`jags.model_options`	a list of optional arguments that are passed to `jags.model()` function.

coda.samples_options
: a list of optional arguments that are passed to coda.samples() function.

verbose
: TRUE or FALSE. If FALSE, no function message and progress bar will be printed.

...
: further arguments. Not currently used.

object
: object to summarize.

x
: object to print

Details

section 2.2.1 and 2.2.2 of the paper listed under reference provides a detailed description of the assumptions and prior distributions of the model.

Note that this package does not include the JAGS library, users need to install JAGS separately. Please check this page for more details: https://sourceforge.net/projects/mcmc-jags/

Value

posterior_sample
: an mcmc.list object generated through the coda.samples() function, which includes posterior samples of the link parameters and response rates generated through the MCMC process

mean_estimate
: BJSM estimate of each parameter:

1. phi1 - lingering effect of the first treatment
2. phi3 - if the patient stays on the same treatment, phi3 is the cumulative effect of stage 1 that occurs on the treatment longer term
3. xi_j - the expected effect of treatment j, j = 1, 2, 3 in the first stage
4. V1,V2 are the variance-covariance matrix of the multivariate distribution. V1 is for patients who stay on the same treatment, and V2 is for patients who switch treatments. This allows those who stay on the same treatment to have a different correlation between stage one stage two outcomes than those who switch treatments.

ci_estimate
: x% credible interval for each parameter. By default round to 2 decimal places, if more decimals are needed, please access the results by [YourResultName]$ci_estimates$CI_low or [YourResultName]$ci_estimates$CI_high

References

Hartman, H., Tamura, R.N., Schipper, M.J. and Kidwell, K.M., 2021. Design and analysis considerations for utilizing a mapping function in a small sample, sequential, multiple assignment, randomized trials with continuous outcomes. Statistics in Medicine, 40(2), pp.312-326. URL: doi:10.1002/sim.8776

Examples

```
trialData <- trialDataMF

BJSM_result <- BJSM_c(
```

```
  data = trialData, xi_prior.mean = c(50, 50, 50),
  xi_prior.sd = c(50, 50, 50), phi3_prior.sd = 20, n_MCMC_chain = 1,
 n.adapt = 1000, MCMC_SAMPLE = 5000, BURIN.IN = 1000, ci = 0.95, n.digits = 5, verbose = FALSE
)

summary(BJSM_result)
print(BJSM_result)
```

`data_binary` *Dataset with binary outcomes*

Description

sample synthetic dataset of snSMART (3 active treatment) with binary outcomes

Usage

```
data_binary
```

Format

This data frame contains the following columns:

treatment_stageI treatment received in stage 1 - possible values: 1 (placebo), 2, 3

response_stageI whether patients respond to stage 1 treatment - possible values: 0 (nonresponder), 1 (responder)

treatment_stageII treatment received in stage 2 - possible values: 2, 3

response_stageII whether patients respond to stage 2 treatment - possible values: 0 (nonresponder), 1 (responder)

Examples

```
mydata <- data_binary
LPJSM_result <- LPJSM_binary(data = mydata, six = TRUE, DTR = TRUE)
```

data_dose *Dose Level dataset with binary outcomes*

Description

sample synthetic dataset of snSMART (dose level treatment) with binary outcomes

Usage

```
data_dose
```

Format

This data frame contains the following columns:

treatment_stageI treatment received in stage 1 - possible values: 1 (placebo), 2, 3

response_stageI whether patients respond to stage 1 treatment - possible values: 0 (nonresponder), 1 (responder)

treatment_stageII treatment received in stage 2 - possible values: 2, 3

response_stageII whether patients respond to stage 2 treatment - possible values: 0 (nonresponder), 1 (responder)

Examples

```
mydata <- data_dose
BJSM_dose_result <- BJSM_binary(
  data = data_dose, prior_dist = c("beta", "gamma"),
  pi_prior = c(3, 17), normal.par = c(0.2, 100), beta_prior = c(2, 2),
  n_MCMC_chain = 2, n.adapt = 100, MCMC_SAMPLE = 2000, ci = 0.95
)
```

groupseqDATA_full *Group sequential full data*

Description

sample synthetic dataset of group sequential trial design snSMART, can be used for final analysis

Usage

```
groupseqDATA_full
```

Format

This data frame contains the following columns:

time.1st.trt first treatment time

time.1st.resp first response time

time.2nd.trt second treatment time

time.2nd.resp second response time

trt.1st treatment arm for first treatment

resp.1st response for first treatment

trt.2nd treatment arm for second treatment

resp.2nd response for second treatment

Examples

```
mydata <- groupseqDATA_full
result2 <- group_seq(
  data = mydata, interim = FALSE, prior_dist = c(
    "beta", "beta", "pareto"
  ), pi_prior = c(0.4, 1.6, 0.4, 1.6, 0.4, 1.6),
  beta_prior = c(1.6, 0.4, 3, 1), MCMC_SAMPLE = 6000, n.adapt = 1000,
  n_MCMC_chain = 1, ci = 0.95, DTR = TRUE
)
```

---

groupseqDATA_look1 *Group sequential data look 1*

Description

sample synthetic dataset of group sequential trial design snSMART, can be used for interim analysis

Usage

```
groupseqDATA_look1
```

Format

This data frame contains the following columns:

time.1st.trt first treatment time

time.1st.resp first response time

time.2nd.trt second treatment time

time.2nd.resp second response time

trt.1st treatment arm for first treatment

resp.1st response for first treatment

trt.2nd treatment arm for second treatment

resp.2nd response for second treatment

Examples

```
mydata <- groupseqDATA_look1

result1 <- group_seq(
  data = mydata, interim = TRUE, drop_threshold_pair = c(0.5, 0.4),
  prior_dist = c("beta", "beta", "pareto"), pi_prior = c(0.4, 1.6, 0.4, 1.6, 0.4, 1.6),
  beta_prior = c(1.6, 0.4, 3, 1), MCMC_SAMPLE = 6000, n.adapt = 1000, n_MCMC_chain = 1
)
```

---

group_seq *BJSM method for interim analysis and final analysis of group sequential trial design*

Description

After obtain real trial data, this function can be used to decide which arm to drop in an interim analysis or provide a full final analysis.

Usage

```
group_seq(
  data,
  interim = TRUE,
  drop_threshold_pair = NULL,
  prior_dist,
  pi_prior,
  beta_prior,
  MCMC_SAMPLE,
  n.adapt,
  thin = 1,
  BURN.IN = 100,
  n_MCMC_chain,
  ci = 0.95,
  DTR = TRUE,
  jags.model_options = NULL,
  coda.samples_options = NULL,
  verbose = FALSE,
  ...
)

## S3 method for class 'summary.group_seq'
print(x, ...)

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

Arguments

data	dataset should include 8 columns: time.1st.trt (first treatment starts time), time.1st.resp (first response time), time.2nd.trt (second treatment starts time), time.2nd.resp (second response time), trt.1st (treatment arm for first treatment), resp.1st (response for first treatment), trt.2nd (treatment arm for second treatment), resp.2nd (response for second treatment) data yet to be observed should be marked as "NA"
interim	indicates whether user is conducting an interim analysis via BJSM (interim = TRUE) or an final analysis via BJSM (interim = FALSE)
drop_threshold_pair	a vector of 2 values (drop_threshold_tau_l, drop_threshold_psi_l). Both drop_threshold_tau_l and drop_threshold_psi_l should be between 0 and 1. only assign value to this parameter when interim = TRUE. See the details section for more explanation
prior_dist	vector of three values ("prior distribution for pi", "prior distribution for beta0", "prior distribution for beta1"), user can choose from "gamma", "beta", "pareto". e.g. prior_dist = c("beta", "beta", "pareto")
pi_prior	vector of six values (a, b, c, d, e, f), where a and b are the parameter a and parameter b of the prior distribution for pi_1A, c and d are the parameter a and parameter b of the prior distribution for pi_1B, and e and f are the parameter a and parameter b of the prior distribution for pi_1C. Please check the Details section for more explanation
beta_prior	vector of four values (beta0_prior.a, beta0_prior.b, beta1_prior.a, beta1_prior.c). beta0_prior.a is the parameter a of the prior distribution for linkage parameter beta0. beta0_prior.b is the parameter b of the prior distribution for linkage parameter beta0. beta1_prior.a is the parameter a of the prior distribution for linkage parameter beta1. beta1_prior.c is the parameter b of the prior distribution for linkage parameter beta1. Please check the Details section for more explanation
MCMC_SAMPLE	number of iterations for MCMC
n.adapt	the number of iterations for adaptation
thin	thinning interval for monitors
BURN.IN	number of burn-in iterations for MCMC
n_MCMC_chain	number of MCMC chains, default to 1
ci	coverage probability for credible intervals, default = 0.95. only assign value to this parameter when interim = FALSE.
DTR	if TRUE, will also return the expected response rate of dynamic treatment regimens. default = TRUE. only assign value to this parameter when interim = FALSE.
jags.model_options	a list of optional arguments that are passed to jags.model() function.
coda.samples_options	a list of optional arguments that are passed to coda.samples() function.

verbose	TRUE or FALSE. If FALSE, no function message and progress bar will be printed.
...	further arguments. Not currently used.
x	object to summarize.

Details

For gamma distribution, prior.a is the shape parameter r, prior.b is the rate parameter lambda. For beta distribution, prior.a is the shape parameter a, prior.b is the shape parameter b. For pareto distribution, prior.a is the scale parameter alpha, prior.b is the shape parameter c (see jags user manual). The individual response rate is regarded as a permanent feature of the treatment. The second stage outcome is modeled conditionally on the first stage results linking the first and second stage response probabilities through linkage parameters.

(paper provided in the reference section, section 2.2.2 Bayesian decision rules. drop_threshold_tau_l and drop_threshold_psi_l correspond to tau_l and psi_l respectively)

Please refer to the paper listed under reference section for detailed definition of parameters. Note that this package does not include the JAGS library, users need to install JAGS separately. Please check this page for more details: https://sourceforge.net/projects/mcmc-jags/

Value

if interim = TRUE, this function returns either 0 - no arm is dropped, or A/B/C - arm A/B/C is dropped

if interim = FALSE, this function returns:

posterior_sample	an mcmc.list object generated through the coda.samples() function, which includes posterior samples of the link parameters and response rates generated through the MCMC process
pi_hat_bjsm	estimate of response rate/treatment effect
se_hat_bjsm	standard error of the response rate
ci_pi_A, ci_pi_B, ci_pi_C	x% credible intervals for treatment A, B, C
diff_AB, diff_BC. diff_AC	estimate of differences between treatments A and B, B and C, A and C
ci_diff_AB, ci_diff_BC, ci_diff_AC	x% credible intervals for the differences between treatments A and B, B and C, A and C
se_AB, se_BC, se_AC	standard error for the differences between treatments A and B, B and C, A and C
beta0_hat, beta1_hat	linkage parameter beta0 and beta1 estimates
se_beta0_hat, se_beta1_hat	standard error of the estimated value of linkage parameter beta0 and beta1

`ci_beta0_hat`, `ci_beta1_hat`
: linkage parameter `beta0` and `beta1` credible interval

`pi_DTR_est`
: expected response rate of dynamic treatment regimens (DTRs)

`pi_DTR_se`
: standard error for the estimated DTR response rate

`ci_pi_AB`, `ci_pi_AC`, `ci_pi_BA`, `ci_pi_BC`, `ci_pi_CA`, `ci_pi_CB`
: x% credible intervals for the estimated DTR response rate

References

Chao, Y.C., Braun, T.M., Tamura, R.N. and Kidwell, K.M., 2020. A Bayesian group sequential small n sequential multiple-assignment randomized trial. Journal of the Royal Statistical Society: Series C (Applied Statistics), 69(3), pp.663-680. URL: doi:10.1111/rssc.12406

Examples

```
mydata <- groupseqDATA_look1

result1 <- group_seq(
  data = mydata, interim = TRUE, drop_threshold_pair = c(0.5, 0.4),
  prior_dist = c("beta", "beta", "pareto"), pi_prior = c(0.4, 1.6, 0.4, 1.6, 0.4, 1.6),
  beta_prior = c(1.6, 0.4, 3, 1), MCMC_SAMPLE = 6000, n.adapt = 1000, n_MCMC_chain = 1
)

summary(result1)


mydata <- groupseqDATA_full
result2 <- group_seq(
  data = mydata, interim = FALSE, prior_dist = c("beta", "beta", "pareto"),
  pi_prior = c(0.4, 1.6, 0.4, 1.6, 0.4, 1.6),
  beta_prior = c(1.6, 0.4, 3, 1), MCMC_SAMPLE = 6000, n.adapt = 1000,
  n_MCMC_chain = 1, ci = 0.95, DTR = TRUE
)

summary(result2)
```

`LPJSM_binary` *LPJSM for snSMART with binary outcomes (3 active treatments or placebo and two dose level)*

Description

A joint-stage regression model (LPJSM) is a frequentist modeling approach that incorporates the responses of both stages as repeated measurements for each subject. Generalized estimating equations (GEE) are used to estimate the response rates of each treatment. The marginal response rates for each DTR can also be obtained based on the GEE results.

Usage

```
LPJSM_binary(data, six = TRUE, DTR = TRUE, ...)

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

## S3 method for class 'summary.LPJSM_binary'
print(x, ...)

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

Arguments

data	dataset with columns named as treatment_stageI, response_stageI, treatment_stageII and response_stageII
six	if TRUE, will run the six beta model, if FALSE will run the two beta model. Default is six = TRUE
DTR	if TRUE, will also return the expected response rate and its standard error of dynamic treatment regimens
...	optional arguments that are passed to geepack::geeglm() function.
object	object to print
x	object to summarize.

Value

a list containing

GEE_output	- original output of the GEE (geeglm) model
pi_hat	- estimate of response rate/treatment effect
sd_pi_hat	- standard error of the response rate
pi_DTR_hat	- expected response rate of dynamic treatment regimens (DTRs)
pi_DTR_se	- standard deviation of DTR estimates

References

Wei, B., Braun, T.M., Tamura, R.N. and Kidwell, K.M., 2018. A Bayesian analysis of small n sequential multiple assignment randomized trials (snSMARTs). Statistics in medicine, 37(26), pp.3723-3732. URL: doi:10.1002/sim.7900

Chao, Y.C., Trachtman, H., Gipson, D.S., Spino, C., Braun, T.M. and Kidwell, K.M., 2020. Dynamic treatment regimens in small n, sequential, multiple assignment, randomized trials: An application in focal segmental glomerulosclerosis. Contemporary clinical trials, 92, p.105989. URL: doi:10.1016/j.cct.2020.105989

Fang, F., Hochstedler, K.A., Tamura, R.N., Braun, T.M. and Kidwell, K.M., 2021. Bayesian methods to compare dose levels with placebo in a small n, sequential, multiple assignment, randomized trial. Statistics in Medicine, 40(4), pp.963-977. URL: doi:10.1002/sim.8813

See Also

BJSM_binary
sample_size

Examples

```
data <- data_binary

LPJSM_result <- LPJSM_binary(data = data, six = TRUE, DTR = TRUE)

summary(LPJSM_result)
```

`sample_size` *Sample size calculation for snSMART with 3 active treatments and a binary outcome*

Description

conduct Bayesian sample size calculation for a snSMART design with 3 active treatments and a binary outcome to distinguish the best treatment from the second-best treatment using the Bayesian joint stage model.

Usage

```
sample_size(pi, beta1, beta0, coverage, power, mu, n, verbose = FALSE)

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

## S3 method for class 'summary.sample_size'
print(x, ...)

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

Arguments

`pi`	a vector with 3 values (`piA`, `piB`, `piC`). `piA` is the the response rate (ranges from 0.01 to 0.99) for treatment A, `piB` is the response rate (ranges from 0.01 to 0.99) for treatment B, `piC` is the response rate (ranges from 0.01 to 0.99) for treatment C
`beta1`	the linkage parameter (ranges from 1.00 to 1/largest response rate) for first stage responders. (A smaller value leads to more conservative sample size calculation because two stages are less correlated)

beta0	the linkage parameter (ranges from 0.01 to 0.99) for first stage non-responders. A larger value leads to a more conservative sample size calculation because two stages are less correlated
coverage	the coverage rate (ranges from 0.01 to 0.99) for the posterior difference of top two treatments
power	the probability (ranges from 0.01 to 0.99) for identify the best treatment
mu	a vector with 3 values (muA, muB, muC). muA is the prior mean (ranges from 0.01 to 0.99) for treatment A, muB is the prior mean (ranges from 0.01 to 0.99) for treatment B, muC is the prior mean (ranges from 0.01 to 0.99) for treatment C
n	a vector with 3 values (nA, nB, nC). nA is the prior sample size (larger than 0) for treatment A. nB is the prior sample size (larger than 0) for treatment B. nC is the prior sample size (larger than 0) for treatment C
verbose	TRUE or FALSE. If FALSE, no function message and progress bar will be printed.
object	object to summarize.
...	further arguments. Not currently used.
x	object to print

Details

Note that this package does not include the JAGS library, users need to install JAGS separately. Please check this page for more details: https://sourceforge.net/projects/mcmc-jags/ This function may take a few minutes to run

Value

final_N	the estimated sample size per arm for this snSMART
critical_value	critical value based on the provided coverage value
grid_result	for each iteration we calculate l, where l belongs to {2 * (pi_(1) - pi_(2)), ..., 0.02, 0.01}; E(D): the mean of the posterior distribution of D, , where D = pi_(1) = pi_(2); Var(D): the variance of the posterior distribution of D; N: the corresponding sample size; and power: the resulting power of this iteration

References

Wei, B., Braun, T.M., Tamura, R.N. and Kidwell, K.M., 2018. A Bayesian analysis of small n sequential multiple assignment randomized trials (snSMARTs). Statistics in medicine, 37(26), pp.3723-3732. URL: doi:10.1002/sim.7900

Wei, B., Braun, T.M., Tamura, R.N. and Kidwell, K., 2020. Sample size determination for Bayesian analysis of small n sequential, multiple assignment, randomized trials (snSMARTs) with three agents. Journal of Biopharmaceutical Statistics, 30(6), pp.1109-1120. URL: doi:10.1080/10543406.2020.1815032

See Also

BJSM_binary

Examples

```
## Not run:
# short running time example
sampleSize <- sample_size(
  pi = c(0.7, 0.5, 0.25), beta1 = 1.4, beta0 = 0.5, coverage = 0.9,
  power = 0.3, mu = c(0.65, 0.55, 0.25), n = c(10, 10, 10)
)

## End(Not run)


sampleSize <- sample_size(
  pi = c(0.7, 0.5, 0.25), beta1 = 1.4, beta0 = 0.5, coverage = 0.9,
  power = 0.8, mu = c(0.65, 0.55, 0.25), n = c(4, 2, 3)
)
```

---

summary.BJSM_binary *Summarizing BJSM fits*

Description

summary method for class "BJSM_binary"

Usage

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

Arguments

object	an object of class "BJSM_binary", usually, a result of a call to BJSM_binary
...	further arguments. Not currently used.

Value

Treatment Effects Estimate a 3 x 5 matrix with columns for the estimated treatment effects, its standard error, coverage probability of its credible interval, lower bound for its credible interval and higher bound for its credible interval

Differences between Treatments a 3 x 5 matrix with columns for the estimated differences in treatment effects between two treatments, its standard error, coverage probability of its credible interval, lower bound and higher bound of the credible interval

Linkage Parameter Estimate a 2 x 5 matrix, if the two beta model is fitted, or a 6 x 5 matrix, if the six beta model is fitted, with columns for the estimated linkage parameters

Expected Response Rate of Dynamic Treatment Regimens (DTR) only when DTR = TRUE

summary.BJSM_dose_binary *Summarizing BJSM fits*

Description

summary method for class BJSM_dose_binary

Usage

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

Arguments

object	an object of class BJSM_dose_binary, usually, a result of a call to BJSM_binary
...	further arguments. Not currently used.

Value

Treatment Effects Estimate a 3 x 5 matrix with columns for the estimated treatment effects, its standard error, coverage probability of its credible interval, lower bound for its credible interval and higher bound for its credible interval

Differences between Treatments a 3 x 5 matrix with columns for the estimated differences in treatment effects between two treatments, its standard error, coverage probability of its credible interval, lower bound and higher bound of the credible interval

Linkage Parameter Estimate a 6 x 5 matrix with columns for the estimated linkage parameters

summary.group_seq *Summarizing BJSM fits*

Description

summary method for class "group_seq"

Usage

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

Arguments

object	an object of class "group_seq", usually, a result of a call to group_seq
...	further arguments. Not currently used.

Value

Treatment Effects Estimate a 3 x 5 matrix with columns for the estimated treatment effects, its standard error, coverage probability of its credible interval, lower bound for its credible interval and higher bound for its credible interval

Differences between Treatments a 3 x 5 matrix with columns for the estimated differences in treatment effects between two treatments, its standard error, coverage probability of its credible interval, lower bound and higher bound of the credible interval

Linkage Parameter Estimate a 2 x 5 matrix, if the two beta model is fitted, or a 6 x 5 matrix, if the six beta model is fitted, with columns for the estimated linkage parameters

Expected Response Rate of Dynamic Treatment Regimens (DTR) only when `DTR = TRUE`

`trialDataMF` *Dataset with continuous outcomes*

Description

sample synthetic dataset of snSMART (mapping function) with continuous outcomes

Usage

```
trialDataMF
```

Format

This data frame contains the following columns:

id participant ID

trt1 treatment received in stage 1 - possible values: 1 (placebo), 2, 3

stage1outcome a number between 0-100 that represents the stage 1 treatment effect

stay indicates whether the participant stayed on the same treatment arm in stage 2 - possible values: 0 (didn't stay), 1 (stayed)

trt2 treatment received in stage 2 - possible values: 2, 3

stage2outcome a number between 0-100 that represents the stage 2 treatment effect

Examples

```
trialData <- trialDataMF

BJSM_result <- BJSM_c(
  data = trialData, xi_prior.mean = c(50, 50, 50),
  xi_prior.sd = c(50, 50, 50), phi3_prior.sd = 20, n_MCMC_chain = 1,
  n.adapt = 1000, MCMC_SAMPLE = 5000, ci = 0.95, n.digits = 5
)

summary(BJSM_result)
print(BJSM_result)
```

Index