

Package ‘prome’

October 8, 2025

Title Patient-Reported Outcome Data Analysis with Stan

Version 2.0.1.9

Description Algorithms and subroutines for patient-reported outcome data analysis.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Biarch true

Depends R (>= 3.5.0), posterior

Imports methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 2.1.1)

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

SystemRequirements GNU make

NeedsCompilation yes

Author Bin Wang [aut, cre] (ORCID: <<https://orcid.org/0000-0002-3689-6932>>)

Maintainer Bin Wang <bwang831@gmail.com>

Repository CRAN

Date/Publication 2025-10-08 12:00:01 UTC

Contents

prome-package	2
bate	2
blinding.lslogit	3
ex100x3	4
MeanHM	4
PropHM	5
xover	5
Index	7

prome-package	<i>The 'prome' package.</i>
---------------	-----------------------------

Description

Algorithms to implement the Bayesian methods to denoise the measurement errors in patient-reported outcome data with repeated measures. Also, two algorithms are included to discount the subgroup means or proportions for clinical studies with multiple subgroups.

bate	<i>Bayesian Hierarchical Model for RPO data with repeated measures</i>
------	--

Description

A Bayesian hierarchical model to denoise PRO data using repeated measures.

Usage

```
bate(x0,x1,group,z,x.range,...)
ResponderAnalysis(x,mcid,type="absolute",conf.level=0.95,show=TRUE)
```

Arguments

x0, x1	Numeric vector/matrix of observations at T0 (baseline) and T1 (end point) of a study.
z	covariates
group	group assignments. Current version support one or two groups only
x.range	range of data 'x0' and 'x1'
x	An R object generated by memixed
mcid	A threshold to define 'responder'
type	The type of responder analysis: absolute or relative changes
conf.level	Confidence level of the credible interval
show	control whether results should be displayed
...	Parameters ("adapt_delta","stepsize","max_treedepth") to improve model fitting/convergence.

Value

- 'xfit': fitted results using stan.
- 'mu.t0': baseline mean.
- 'sig.t0': baseline SD.
- 'sig.me': SD of measurement errors.
- 'mu.active': mean effect size of active treatment.
- 'sig.active': sd of effect size of active treatment.
- 'mu.sham': mean effect size of sham treatment.
- 'sig.sham': sd of effect size of sham treatment.

Examples

```

data(n100x3)
out1 <- bate(x0=ex100x3$w0,x1=ex100x3$w1,group=ex100x3$group)
out1
ResponderAnalysis(out1,mcid=1,type="abs")
out2 <- bate(x0=ex100x3$w0,x1=ex100x3$w1,group=ex100x3$group,
  control = list(adapt_delta = 0.8,
    stepsize = 5,
    max_treedepth = 10)
)
out2
ResponderAnalysis(out2,mcid=1,type="abs")
out <- out2
ResponderAnalysis(out,mcid=0.5,type="abs")
ResponderAnalysis(out,mcid=1,type="abs")
ResponderAnalysis(out,mcid=1.5,type="abs")
ResponderAnalysis(out,mcid=0.3,type="relative")
ResponderAnalysis(out,mcid=0.2,type="relative")
ResponderAnalysis(out,mcid=0.1,type="relative")

```

blinding.lslogit

*Latent Shift Logistic Regression***Description**

To be updated.

Usage

```
blinding.lslogit(x, group, guess, mu0 = 0, s0 = 1, ...)
```

Arguments

x, guess	outcome variable and guess response from blinding survey
group	group assignments. Current version support one or two groups only
mu0, s0	initial mean and sd of the latent variable of having sham effects
...	Parameters ("adapt_delta","stepsize","max_treedepth") to improve model fitting/convergence.

Value

- ‘sig.sham’: sd of effect size of sham treatment.

Examples

```

guess <- rbinom(100,1,0.5)
group <- rbinom(100,1,0.5)
x <- rnorm(100,3,1)
blinding.lslogit(x=x,group=group,guess=guess)

```

ex100x3	<i>Sample PRO Data With Repeated Measures</i>
---------	---

Description

A simulated data set of patient-reported outcomes with repeated measures.

Format

A data frame with observations at baseline and at a follow-up time.

w0	matrix	measures at baseline
w1	matrix	measures at follow-up time
group	character	group assignment

MeanHM	<i>Bayesian Hierarchical Model for Information Borrowing for Means</i>
--------	--

Description

To compute the mean values of subgroups based on a Bayesian hierarchical model.

Usage

MeanHM(x,sigma)

Arguments

x	Numeric vector of observations for the subgroups.
sigma	hyper-parameter. to be estimated or can be given.

Value

- ‘theta’: population mean.
- ‘sigma’: population standard deviation.

Examples

```
x1 <- rnorm(100,2,1)
x2 <- rnorm(100,3,1.5)
x3 <- rnorm(100,4,1.9)
x <- cbind(x1,x2,x3)
MeanHM(x,sigma=0.5)
```

PropHM	<i>Bayesian Hierarchical Model for Information Borrowing for Proportions</i>
--------	--

Description

To compute the proportions of the subgroups assuming the subgroups follow the same binomial distribution with parameter p . The approach on partial pooling by Bob Carpenter has been used – "Hierarchical Partial Pooling for Repeated Binary Trials" <https://mc-stan.org/users/documentation/case-studies/pool-binary-trials.html>

Usage

```
PropHM(x, n, kappa)
```

Arguments

x	Numeric vector of events.
n	Numeric vector of group sample sizes.
kappa	$\text{kappa} = \alpha + \beta > 1$. Must be given if the number of subgroups is 2.

Value

- 'data': data with estimates.
- 'alpha': parameter of the beta distribution.
- 'beta': parameter of the beta distribution.

Examples

```
out <- PropHM(x=c(5,10,2),n=c(20,50,30))
```

xover	<i>Bayesian analysis of 2x2 crossover trial data</i>
-------	--

Description

A Bayesian hierarchical model to analysis data from 2x2 (AB/BA) crossover trials.

Usage

```
xover(group,y1,y2,y0,...)
```

Arguments

`y0, y1, y2` vectors of data from baseline, period 1, and period 2, respectively.
`group` group or treatment sequence.
`...` other parameters, i.e. 'control' for model fitting.

Value

- 'stat': summary statistics.
- 'best': estimates using Bayesian analysis.

Examples

```
xover(y0=rnorm(20,34,1.5),y1=rnorm(20,30,2),  
      y2=rnorm(20,25,1.5),group=round(runif(20)<0.5))
```

Index

* datasets

ex100x3, [4](#)

bate, [2](#)

blinding.lslogit, [3](#)

ex100x3, [4](#)

FitLogitStan (blinding.lslogit), [3](#)

MeanHM, [4](#)

n100x3 (ex100x3), [4](#)

plot.memix (bate), [2](#)

pratio (bate), [2](#)

print.memix (bate), [2](#)

print.xover (xover), [5](#)

prome (prome-package), [2](#)

prome-package, [2](#)

PropHM, [5](#)

ResponderAnalysis (bate), [2](#)

xover, [5](#)