

Package ‘dlmtree’

May 8, 2026

Type Package

Title Bayesian Treed Distributed Lag Models

Version 1.1.1

Description Estimation of distributed lag models (DLMs) based on a Bayesian additive regression trees framework. Includes several extensions of DLMs: treed DLMs and distributed lag mixture models (Mork and Wilson, 2023) <[doi:10.1111/biom.13568](https://doi.org/10.1111/biom.13568)>; treed distributed lag nonlinear models (Mork and Wilson, 2022) <[doi:10.1093/biostatistics/kxaa051](https://doi.org/10.1093/biostatistics/kxaa051)>; heterogeneous DLMs (Mork, et. al., 2024) <[doi:10.1080/01621459.2023.2258595](https://doi.org/10.1080/01621459.2023.2258595)>; monotone DLMs (Mork and Wilson, 2024) <[doi:10.1214/23-BA1412](https://doi.org/10.1214/23-BA1412)>. The package also includes visualization tools and a 'shiny' interface to check model convergence and to help interpret results.

License GPL (>= 3)

Encoding UTF-8

LazyData false

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.4), dplyr, ggplot2, shiny, shinythemes, tidyr, mgcv, ggridges, coda

LinkingTo Rcpp, RcppArmadillo, RcppEigen

RoxygenNote 7.3.2

URL <https://github.com/danielmork/dlmtree>,
<https://danielmork.github.io/dlmtree/>

BugReports <https://github.com/danielmork/dlmtree/issues>

NeedsCompilation yes

Author Daniel Mork [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-7924-0706>>),
Seongwon Im [aut] (ORCID: <<https://orcid.org/0009-0000-8447-5852>>),
Ander Wilson [aut] (ORCID: <<https://orcid.org/0000-0003-4774-3883>>)

Maintainer Daniel Mork <dmork@hsph.harvard.edu>

Repository CRAN

Date/Publication 2026-02-03 03:20:02 UTC

Contents

adj_coexposure	3
coExp	4
combine.models	5
combine.models.tdlmm	5
cppIntersection	6
diagnose.summary.hdlm	6
dImEst	7
dlmtree	8
dlmtree.control.diagnose	11
dlmtree.control.family	12
dlmtree.control.het	13
dlmtree.control.hyper	14
dlmtree.control.mcmc	14
dlmtree.control.mix	15
dlmtree.control.monotone	15
dlmtree.control.tdlm	16
dlmtreeGPFixedGaussian	17
dlmtreeGPGaussian	17
dlmtreeHDLMGaussian	18
dlmtreeHDLMMGaussian	18
dlmtreeTDLMMFixedGaussian	19
dlmtreeTDLMNestedGaussian	19
dlmtreeTDLM_cpp	20
dlmEst	20
dlmPLEst	21
drawTree	21
estDLM	22
exposureCov	23
get_sbd_dlmtree	23
mixEst	25
monotdlm_Cpp	25
pip	26
plot.summary.monotone	27
plot.summary.tdlm	27
plot.summary.tdlmm	28
plot.summary.tdlm	29
pm25Exposures	30
ppRange	30
predict	31
print	33
rcpp_pgdraw	34
rtmvnorm	34
ruleIdx	35
scaleModelMatrix	35
shiny	36
sim.hdlmm	36

sim.tdlmm	38
sim.tdlm	39
splitPIP	40
splitpoints	40
summary	41
tdlmm_Cpp	43
tdlm_Cpp	43
zeroToInfNormCDF	44
zinbCo	44

Index	45
--------------	-----------

adj_coexposure	<i>Adjusting for expected changes in co-exposure (TDLMM)</i>
----------------	--

Description

Estimates the marginal effects of an exposure while accounting for expected changes in co-occurring exposures at the same time point. Values of co-occurring exposures are modeled nonlinearly using a spline model with predictions made at the lower and upper values for the exposure of interest.

Usage

```
adj_coexposure(
  exposure.data,
  object,
  contrast_perc = c(0.25, 0.75),
  contrast_exp = list(),
  conf.level = 0.95,
  keep.mcmc = FALSE,
  verbose = TRUE
)
```

Arguments

exposure.data	Named list of exposure matrices used as input to TDLMM.
object	Model output for TDLMM from dlmtree() function.
contrast_perc	2-length vector of percentiles or named list corresponding to lower and upper exposure percentiles of interest. Names must equal list names in 'exposure.data'.
contrast_exp	Named list consisting lower and upper exposure values. This takes precedence over contrast_perc if both inputs are used.
conf.level	Confidence level used for estimating credible intervals. Default is 0.95.
keep.mcmc	If TRUE, return posterior samples.
verbose	TRUE (default) or FALSE: print output

Details

adj_coexposure

Value

A list with the following components (or posterior samples if keep.mcmc = TRUE):

Name	vector of exposure names
Time	integer vector of lags
Effect	posterior mean of marginal effects
SE	standard error of the estimate
Lower	lower bound of credible interval of the marginal effect estimate
Upper	upper bound of credible interval of the marginal effect estimate
cEffect	cumulative marginal effects
cLower	lower bound of credible interval of the cumulative marginal effect
cUpper	upper bound of credible interval of the cumulative marginal effect
CW	boolean vector indicating critical window

 coExp

Randomly sampled exposure from Colorado counties

Description

Matrix of five different exposures, each measured over 40 weeks.

Usage

```
data(coExp)
```

Format

matrix

Source

https://aqs.epa.gov/aqsweb/airdata/download_files.html

References

<https://www.epa.gov/outdoor-air-quality-data>

combine.models	<i>Combines information from DLMs of single exposure</i>
----------------	--

Description

Method for combining information from DLMs of single exposure

Usage

```
combine.models(mlist)
```

Arguments

`mlist` a list of models

Details

combine.models

Value

A data frame with model fit information of the models included in the list

combine.models.tdlmm	<i>Combines information from DLMs of mixture exposures.</i>
----------------------	---

Description

Method for combining information from DLMs of mixture exposures.

Usage

```
combine.models.tdlmm(mlist)
```

Arguments

`mlist` a list of models

Details

combine.models.tdlmm

Value

A data frame with model fit information of the models included in the list

cppIntersection *fast set intersection tool assumes sorted vectors A and B*

Description

fast set intersection tool assumes sorted vectors A and B

Usage

```
cppIntersection(A, B)
```

Arguments

A	sorted integer vector A
B	sorted integer vector B

Value

vector of resulting intersection

diagnose.summary.hdlm *diagnose*

Description

diagnose generic function for S3method

Usage

```
## S3 method for class 'summary.hdlm'  
diagnose(x, ...)  
  
## S3 method for class 'summary.hdlmm'  
diagnose(x, ...)  
  
## S3 method for class 'summary.monotone'  
diagnose(x, ...)  
  
## S3 method for class 'summary.tdlm'  
diagnose(x, ...)  
  
## S3 method for class 'summary.tdlmm'  
diagnose(x, ...)  
  
## S3 method for class 'summary.tdlmm'  
diagnose(x, ...)
```

```
diagnose(x, ...)
```

```
diagnose(x, ...)
```

Arguments

`x` a summary object resulting from `summary()` applied to an object of class `'tdlm'`, `'tdlmm'`, `'tdlnm'`, `'hdlm'`, `'hdlmm'`, `'monotone'`

`...` not used.

Value

shiny interface for assessing model convergence. The interface includes tabs for MCMC diagnostics such as trace plots, density plots, and convergence measures for distributed lag effects, DLM tree sizes, and hyperparameters.

dLmEst	<i>Calculates the distributed lag effect with DLM matrix for linear models.</i>
--------	---

Description

Calculates the distributed lag effect with DLM matrix for linear models.

Usage

```
dLmEst(dlm, nlags, nsamp)
```

Arguments

`dlm` A numeric matrix containing the model fit information

`nlags` total number of lags

`nsamp` number of mcmc iterations

Value

A cube object of lag effect x lag x mcmc

dlmtree

*Fit tree structured distributed lag models***Description**

The 'dlmtree' function accommodates various response variable types, including continuous, binary, and zero-inflated count values. The function is designed to handle both single exposure and exposure mixtures. For a single exposure, users are offered options to model non-linear effects (tdlnm), linear effects (tdlm), or heterogeneous subgroup/individualized effects (hdlm). In the case of exposure mixtures, the function supports lagged interactions (tdlmm), and heterogeneous subgroup/individualized effects (hdlmm) allowing for a comprehensive exploration of mixture exposure heterogeneity. Additionally, users can fine-tune parameters to impose effect shrinkage and perform exposure selection, enhancing the adaptability and precision of the modeling process. For more detailed documentation, visit: [dlmtree website](#).

Usage

```
dlmtree(
  formula,
  data,
  exposure.data,
  dlm.type = "linear",
  family = "gaussian",
  mixture = FALSE,
  het = FALSE,
  control.mcmc = list(),
  control.hyper = list(),
  control.family = list(),
  control.tdlm = list(),
  control.het = list(),
  control.mix = list(),
  control.monotone = list(),
  control.diagnose = list()
)
```

Arguments

formula	object of class formula, a symbolic description of the fixed effect model to be fitted, e.g. $y \sim a + b$.
data	data frame containing variables used in the formula.
exposure.data	numerical matrix of exposure data with same length as data, for a mixture setting (tdlmm, hdlmm): named list containing equally sized numerical matrices of exposure data having same length as data.
dlm.type	dlm model specification: "linear" (default), "nonlinear", "monotone".
family	'gaussian' for continuous response, 'logit' for binomial, 'zinb' for zero-inflated negative binomial.

mixture	flag for mixture, set to TRUE for tdlmm and hdlmm (default: FALSE).
het	flag for heterogeneity, set to TRUE for hdlm and hdlmm (default: FALSE).
control.mcmc	list of MCMC control parameters. This is passed to dltree.control.mcmc .
control.hyper	list of hyperparameter control parameters. This is passed to dltree.control.hyper
control.family	list of family control parameters. This is passed to dltree.control.family
control.tdlm	list of TDLNM control parameters. This is passed to dltree.control.tdlm
control.het	list of control parameters for heterogeneous models. This is passed to dltree.control.het
control.mix	list of mixture control parameters. This is passed to dltree.control.mix
control.monotone	list of control parameters for monotone model. This is passed to dltree.control.monotone
control.diagnose	list of control parameters for diagnostics. This is passed to dltree.control.diagnose

Details

dltree

Model is recommended to be run for at minimum 5000 burn-in iterations followed by 15000 sampling iterations with a thinning factor of 5. Convergence can be checked by re-running the model and validating consistency of results. Examples are provided below for the syntax for running different types of models. For more examples, visit: [dltree website](#).

Value

object of one of the classes: tdlm, tdlmm, tdlm, hdlm, hdlmm, monotone

Examples

```
# The first three examples are for one lagged exposure

# treed distributed lag model (TDLM)
# binary outcome with logit link

D <- sim.tdlmm(sim = "A", mean.p = 0.5, n = 1000)
tdlm.fit <- dltree(y ~ .,
                  data = D$dat,
                  exposure.data = D$exposures[[1]],
                  dlm.type = "linear",
                  family = "logit",
                  control.family = list(binomial.size = 1))

# summarize results
tdlm.sum <- summary(tdlm.fit)
tdlm.sum

# plot results
plot(tdlm.sum)
```

```

# Treed distributed lag nonlinear model (TDLNM)
# Gaussian regression model
D <- sim.tdlm(sim = "A", error.to.signal = 1)
tdlnm.fit <- dlmtree(formula = y ~ .,
                     data = D$dat,
                     exposure.data = D$exposures,
                     dlm.type = "nonlinear",
                     family = "gaussian")

# summarize results
tdlnm.sum <- summary(tdlnm.fit)
tdlnm.sum

# plot results
plot(tdlnm.sum)

# Heterogeneous TDLM (HDLM), similar to first example but with heterogeneous exposure response
D <- sim.hdlm(sim = "B", n = 1000)
hdlm.fit <- dlmtree(y ~ .,
                   data = D$dat,
                   exposure.data = D$exposures,
                   dlm.type = "linear",
                   family = "gaussian",
                   het = TRUE)

# summarize results
hdlm.sum <- summary(hdlm.fit)
hdlm.sum

# shiny app for HDLM
if (interactive()) {
  shiny(hdlm.fit)
}

# The next two examples are for a mixture (or multivariate) exposure

# Treed distributed lag mixture model (TDLMM)
# Model for mixture (or multivariate) lagged exposures
# with a homogenous exposure-time-response function
D <- sim.tdlm(sim = "B", error = 25, n = 1000)
tdlmm.fit <- dlmtree(y ~ .,
                    data = D$dat, exposure.data = D$exposures,
                    control.mix = list(interactions = "noself"),
                    dlm.type = "linear", family = "gaussian",
                    mixture = TRUE)

```

```
# summarize results
tdlmm.sum <- summary(tdlmm.fit)

# plot the marginal exposure-response for one exposure
plot(tdlmm.sum, exposure1 = "e1")

# plot exposure-response surface
plot(tdlmm.sum, exposure1 = "e1", exposure2 = "e2")

# heterogeneous version of TDLMM
D <- sim.hdlmm(sim = "D", n = 1000)
hdlmm.fit <- dltree(y ~ .,
                   data = D$dat,
                   exposure.data = D$exposures,
                   dlm.type = "linear",
                   family = "gaussian",
                   mixture = TRUE,
                   het = TRUE)

# summarize results
hdlmm.sum <- summary(hdlmm.fit)
hdlmm.sum

# summarize results
if (interactive()) {
  shiny(hdlmm.fit)
}
```

dltree.control.diagnose

Diagnostic control settings for dltree model fitting

Description

Diagnostic control settings for dltree model fitting

Usage

```
dltree.control.diagnose(
  subset = NULL,
  lowmem = FALSE,
  verbose = TRUE,
  save.data = TRUE,
  diagnostics = FALSE,
```

```

    initial.params = NULL
  )

```

Arguments

subset	integer vector to analyze only a subset of data and exposures.
lowmem	TRUE or FALSE (default): turn on memory saver for DLNM, slower computation time.
verbose	TRUE (default) or FALSE: print output
save.data	TRUE (default) or FALSE: save data used for model fitting. This must be set to TRUE to use shiny() function on hdlm or hdlmm
diagnostics	TRUE or FALSE (default) keep model diagnostic such as the number of terminal nodes and acceptance ratio.
initial.params	initial parameters for fixed effects model, FALSE = none (default), "glm" = generate using GLM, or user defined, length must equal number of parameters in fixed effects model.

Value

list of control parameters for diagnostics.

dlmtree.control.family

Family control settings for dlmtree model fitting

Description

Family control settings for dlmtree model fitting

Usage

```
dlmtree.control.family(binomial.size = 1, formula.zi = NULL)
```

Arguments

binomial.size	integer type scalar (if all equal, default: 1) or vector defining binomial size for 'logit' family.
formula.zi	(only applies to family = 'zinb') object of class formula, a symbolic description of the fixed effect of zero-inflated (ZI) model to be fitted, e.g. $y \sim a + b$. This only applies to ZINB where covariates for ZI model are different from NB model. This is set to the argument 'formula' by default.

Value

list of family control parameters.

dltree.control.het *Control settings for dltree model fitting, when used for heterogeneous models*

Description

Control settings for dltree model fitting, when used for heterogeneous models

Usage

```
dltree.control.het(  
  modifiers = "all",  
  modifier.splits = 20,  
  modtree.params = c(0.95, 2),  
  modtree.step.prob = c(0.25, 0.25, 0.25),  
  dltree.type = "shared",  
  selection.prior = 0.5  
)
```

Arguments

modifiers string vector containing desired modifiers to be included in a modifier tree. The strings in the vector must match the names of the columns of the data. By default, a modifier tree considers all covariates in the formula as modifiers unless stated otherwise.

modifier.splits integer value to determine the possible number of splitting points that will be used for a modifier tree.

modtree.params numerical vector of alpha and beta hyperparameters controlling modifier tree depth. (default: alpha = 0.95, beta = 2)

modtree.step.prob numerical vector for probability of each step for modifier tree updates: 1) grow, 2) prune, 3) change. (default: c(0.25, 0.25, 0.25))

dltree.type specification of dltree type for HDLM: shared (default) or nested.

selection.prior scalar hyperparameter for sparsity of modifiers. Must be between 0.5 and 1. Smaller value corresponds to increased sparsity of modifiers.

Value

list of control parameters for heterogeneous models.

dlmtree.control.hyper *Hyperparameter control settings for dlmtree model fitting*

Description

Hyperparameter control settings for dlmtree model fitting

Usage

```
dlmtree.control.hyper(
  shrinkage = "all",
  params = c(0.95, 2),
  step.prob = c(0.25, 0.25)
)
```

Arguments

shrinkage	character "all" (default), "trees", "exposures", "none", turns on horseshoe-like shrinkage priors for different parts of model.
params	numerical vector of alpha and beta hyperparameters controlling dlm tree depth. (default: alpha = 0.95, beta = 2)
step.prob	numerical vector for probability of each step for dlm tree updates: 1) grow/prune, 2) change, 3) switch exposure. (default: c(0.25, 0.25, 0.25))

Value

list of hyperparameter control parameters.

dlmtree.control.mcmc *MCMC control settings for dlmtree model fitting*

Description

MCMC control settings for dlmtree model fitting

Usage

```
dlmtree.control.mcmc(n.trees = 20, n.burn = 1000, n.iter = 2000, n.thin = 10)
```

Arguments

n.trees	integer for number of trees in ensemble.
n.burn	integer for length of MCMC burn-in.
n.iter	integer for number of MCMC iterations to run model after burn-in.
n.thin	integer MCMC thinning factor, i.e. keep every tenth iteration.

Value

list of MCMC control parameters.

dlmtree.control.mix	<i>Control settings for dlmtree model fitting, when used for mixture models</i>
---------------------	---

Description

Control settings for dlmtree model fitting, when used for mixture models

Usage

```
dlmtree.control.mix(interactions = "noself", sparsity.prior = 1)
```

Arguments

`interactions` 'noself' (default) which estimates interactions only between two different exposures, 'all' which also allows interactions within the same exposure, or 'none' which eliminates all interactions and estimates only main effects of each exposure.

`sparsity.prior` positive scalar hyperparameter for sparsity of exposures. (default: 1)

Value

list of mixture control parameters.

dlmtree.control.monotone	<i>Control settings for dlmtree model fitting, when used for monotone model</i>
--------------------------	---

Description

Control settings for dlmtree model fitting, when used for monotone model

Usage

```
dlmtree.control.monotone(
  gamma0 = NULL,
  sigma = NULL,
  tree.time.params = c(0.95, 2),
  tree.exp.params = c(0.95, 2),
  time.kappa = NULL
)
```

Arguments

<code>gamma0</code>	vector (with length equal to number of lags) of means for logit-transformed prior probability of split at each lag; e.g., <code>gamma_0l = 0</code> implies mean prior probability of split at lag $l = 0.5$.
<code>sigma</code>	symmetric matrix (usually with only diagonal elements) corresponding to <code>gamma_0</code> to define variances on prior probability of split; e.g., <code>gamma_0l = 0</code> with l th diagonal element of <code>sigma=2.701</code> implies that 95% of the time the prior probability of split is between 0.005 and 0.995, as a second example setting <code>gamma_0l=4.119</code> and the corresponding diagonal element of <code>sigma=0.599</code> implies that 95% of the time the prior probability of a split is between 0.8 and 0.99.
<code>tree.time.params</code>	numerical vector of hyperparameters for monotone time tree.
<code>tree.exp.params</code>	numerical vector of hyperparameters for monotone exposure tree.
<code>time.kappa</code>	scaling factor in dirichlet prior that goes alongside ‘ <code>time.split.prob</code> ’ to control the amount of prior information given to the model for deciding probabilities of splits between adjacent lags.

Value

list of control parameters for monotone model.

`dlmtree.control.tdlnm` *Control settings for dlmtree model fitting, when used for TDLNM*

Description

Control settings for dlmtree model fitting, when used for TDLNM

Usage

```
dlmtree.control.tdlnm(
  exposure.splits = 20,
  time.split.prob = NULL,
  exposure.se = NULL
)
```

Arguments

`exposure.splits`
 scalar indicating the number of splits (divided evenly across quantiles of the exposure data) or list with two components: ‘`type`’ = ‘`values`’ or ‘`quantiles`’, and ‘`split.vals`’ = a numerical vector indicating the corresponding exposure values or quantiles for splits.

time.split.prob	probability vector of a splitting probabilities for time lags. (default: uniform probabilities)
exposure.se	numerical matrix of exposure standard errors with same size as exposure.data or a scalar smoothing factor representing a uniform smoothing factor applied to each exposure measurement. (default: sd(exposure.data)/2)

Value

list of TDLNM control parameters.

dlmtreeGPFixedGaussian

dlmtree model with fixed Gaussian process approach

Description

dlmtree model with fixed Gaussian process approach

Usage

dlmtreeGPFixedGaussian(model)

Arguments

model A list of parameter and data contained for the model fitting

Value

A list of dlmtree model fit, mainly posterior mcmc samples

dlmtreeGPGaussian

dlmtree model with Gaussian process approach

Description

dlmtree model with Gaussian process approach

Usage

dlmtreeGPGaussian(model)

Arguments

model A list of parameter and data contained for the model fitting

Value

A list of dlmtree model fit, mainly posterior mcmc samples

`dltreeHDLMGaussian` *dltree model with shared HDLM approach*

Description

dltree model with shared HDLM approach

Usage

```
dltreeHDLMGaussian(model)
```

Arguments

`model` A list of parameter and data contained for the model fitting

Value

A list of dltree model fit, mainly posterior mcmc samples

`dltreeHDLMMGaussian` *dltree model with HDLMM approach*

Description

dltree model with HDLMM approach

Usage

```
dltreeHDLMMGaussian(model)
```

Arguments

`model` A list of parameter and data contained for the model fitting

Value

A list of dltree model fit, mainly posterior mcmc samples

`dltreeTDLMLFixedGaussian`*dltree model with fixed Gaussian approach*

Description

dltree model with fixed Gaussian approach

Usage

```
dltreeTDLMLFixedGaussian(model)
```

Arguments

model A list of parameter and data contained for the model fitting

Value

A list of dltree model fit, mainly posterior mcmc samples

`dltreeTDLMLNestedGaussian`*dltree model with nested Gaussian approach*

Description

dltree model with nested Gaussian approach

Usage

```
dltreeTDLMLNestedGaussian(model)
```

Arguments

model A list of parameter and data contained for the model fitting

Value

A list of dltree model fit, mainly posterior mcmc samples

dlnmtreeTDLm_cpp	<i>dlnmtree model with nested HDLM approach</i>
------------------	---

Description

dlnmtree model with nested HDLM approach

Usage

```
dlnmtreeTDLm_cpp(model)
```

Arguments

model	A list of parameter and data contained for the model fitting
-------	--

Value

A list of dlnmtree model fit, mainly posterior mcmc samples

dlnmEst	<i>Calculates the distributed lag effect with DLM matrix for non-linear models.</i>
---------	---

Description

Calculates the distributed lag effect with DLM matrix for non-linear models.

Usage

```
dlnmEst(dlnm, predAt, nlags, nsamp, center, se)
```

Arguments

dlnm	A numeric matrix containing the model fit information
predAt	Number of splits in the model
nlags	total number of lags
nsamp	number of mcmc iterations
center	center parameter
se	Standard error parameter

Value

A cube object of lag effect x lag x mcmc

dlnmPLEst	<i>Calculates the distributed lag effect with DLM matrix for non-linear models.</i>
-----------	---

Description

Calculates the distributed lag effect with DLM matrix for non-linear models.

Usage

```
dlnmPLEst(dlnm, predAt, nlags, nsamp, center)
```

Arguments

dlnm	A numeric matrix containing the model fit information
predAt	Number of splits in the model
nlags	total number of lags
nsamp	number of mcmc iterations
center	center parameter

Value

A cube object of lag effect x lag x mcmc

drawTree	<i>Draws a new tree structure</i>
----------	-----------------------------------

Description

A recursive method for drawing a new tree structure

Usage

```
drawTree(depth, alpha, beta)
```

Arguments

depth	depth of a tree
alpha	tree shape parameter, $0 < \alpha < 1$
beta	tree size parameter, $\beta > 0$

Details

drawTree

Value

integer value of number of terminal nodes

estDLM

Calculates subgroup-specific lag effects for heterogeneous models

Description

Method for calculating subgroup-specific lag effects for heterogeneous models: HDLM, HDLMM

Usage

```
estDLM(
  object,
  new.data,
  group.index,
  conf.level = 0.95,
  exposure = NULL,
  keep.mcmc = FALSE,
  mem.safe = FALSE,
  verbose = TRUE
)
```

Arguments

object	object of a model fit. Must be 'hdlm' or 'hdlmm'
new.data	data frame with new observations with the same number of modifiers
group.index	list of index (row numbers) for subgroup specification
conf.level	confidence level for credible interval of effects
exposure	exposure of interest for 'hdlmm' method
keep.mcmc	store mcmc in the output
mem.safe	boolean memory parameter for rule index
verbose	TRUE (default) or FALSE: print output

Details

estDLM

Value

A list with the following components:

conf.level	Specified confidence level
mod	a list of modifiers with a vector of values from the model
n	Number of observation per specified subgroup

groupIndex	list of index (row numbers) for specified subgroup
d1mMean	distributed lag effects per subgroups
d1mCI	credible intervals for distributed lag effects per subgroups
d1mCum	cumulative effects per subgroups
d1Function	type of DLM class
plotData	data frame built for easier visualization of distributed lag effects for each subgroup (facet)

exposureCov	<i>Exposure covariance structure</i>
-------------	--------------------------------------

Description

Matrix containing pairwise covariances for real exposure data consisting of five different exposures, each measured over 37 weeks.

Usage

```
data(exposureCov)
```

Format

matrix

Source

https://aqs.epa.gov/aqsweb/airdata/download_files.html

References

<https://www.epa.gov/outdoor-air-quality-data>

get_sbd_d1mtree	<i>Download simulated data for d1mtree articles</i>
-----------------	---

Description

Download simulated data for d1mtree articles

Usage

```
get_sbd_d1mtree()
```

Value

A data frame with 10000 rows (observations) and 202 variables. All data is simulated. The variables are:

bwgaz	Outcome to be used. Simulated birth weight for gestational age z-score.
ChildSex	Binary sex of child.
MomAge	Continuous age in years.
GestAge	Continuous estimated gestational age at birth in weeks.
MomHeightIn	Continuous maternal height in inches.
MomPriorWeightLbs	Continuous mothers pre-pregnancy weight in pounds.
MomPriorBMI	Continuous mothers pre-pregnancy BMI.
race	Categorical race.
Hispanic	Binary indicator of Hispanic.
MomEdu	Categorical maternal highest educational attainment.
SmkAny	Binary indicator of any smoking during pregnancy.
Marital	Categorical maternal marital status.
Income	Categorical income.
EstDateConcept	Estimated date of conception.
EstMonthConcept	Estimated month of conception.
EstYearConcept	Estimated year of conception.
pm25_1 - pm25_37	Weekly average exposure to PM2.5 for weeks 1 to 37. The columns are already scaled by the exposure IQR of 0.35.
no2_1 - no2_37	Weekly average exposure to NO2 for weeks 1 to 37. The columns are already scaled by the exposure IQR of 9.13.
so2_1 - so2_37	Weekly average exposure to SO2 for weeks 1 to 37. The columns are already scaled by the exposure IQR of 0.96.
co2_1 - co2_37	Weekly average exposure to CO for weeks 1 to 37. The columns are already scaled by the exposure IQR of 0.15.
temp_1 - temp_37	Weekly average exposure to temperature for weeks 1 to 37. The columns are already scaled by the exposure IQR of 27.93
source	Variable indicating that the data came from the bdlim package.

Examples

```
sbd_dlmtree <- get_sbd_dlmtree()
```

mixEst	<i>Calculates the lagged interaction effects with MIX matrix for linear models.</i>
--------	---

Description

Calculates the lagged interaction effects with MIX matrix for linear models.

Usage

```
mixEst(dlm, nlags, nsamp)
```

Arguments

dlm	A numeric matrix containing the model fit information
nlags	total number of lags
nsamp	number of mcmc iterations

Value

A cube object of interaction effect x lag x mcmc

monotdlnm_Cpp	<i>dlmtree model with monotone tdlm approach</i>
---------------	--

Description

dlmtree model with monotone tdlm approach

Usage

```
monotdlnm_Cpp(model)
```

Arguments

model	A list of parameter and data contained for the model fitting
-------	--

Value

A list of dlmtree model fit, mainly posterior mcmc samples

pip *Calculates posterior inclusion probabilities (PIPs) for modifiers in HDLM & HDLMM*

Description

Method for calculating posterior inclusion probabilities (PIPs) for modifiers in HDLM & HDLMM

Usage

```
pip(object, type = 1)
```

Arguments

object	An object of class dlmtree.
type	Type=1 indicates single modifier PIPs. Type=2 indicates joint modifier PIPs for two modifiers.

Details

pip

Value

numeric vector of PIPs named with modifiers (type=1) or data.frame of PIPs with the following columns (type=2):

var1	first modifier of joint modifiers
var2	second modifier of joint modifiers
pip	joint PIPs for the two modifiers

Examples

```
# Posterior inclusion probability with HDLM
D <- sim.hdlmm(sim = "B", n = 1000)
fit <- dlmtree(y ~ .,
               data = D$dat,
               exposure.data = D$exposures,
               dlm.type = "linear",
               family = "gaussian",
               het = TRUE)

pip(fit)
pip(fit, type = 2)
```

plot.summary.monotone *Returns variety of plots for model summary of class 'monotone'*

Description

Method for returning variety of plots for model summary of class 'monotone'

Usage

```
## S3 method for class 'summary.monotone'
plot(x, plot.type = "mean", val = c(), time = c(), ...)
```

Arguments

x	object of class 'summary.monotone', output of summary of 'monotone'
plot.type	string indicating plot type, options are 'mean' (default) which shows mean exposure-time response surface, 'se', 'ci-min', 'ci-max', 'slice' which takes a slice of the plot at a given 'val' or 'time', 'animate' which creates a animation of slices of the surface plot across exposure values (requires package ganimate)
val	exposure value for slice plot
time	time value for slice plot
...	additional parameters to alter plots: 'main', 'xlab', 'ylab', 'flab' which sets the effect label for surface plots, 'start.time' which sets the first time value

Details

plot.summary.monotone

Value

A plot of distributed lag effect estimated with monotone-TDLNM

plot.summary.tdlm *Plots a distributed lag function for model summary of 'tdlm'*

Description

Method for plotting a distributed lag function for model summary of 'tdlm'

Usage

```
## S3 method for class 'summary.tdlm'
plot(x, ...)
```

Arguments

x object of class 'summary.tdlm', output of summary of 'tdlm'
 ... additional plotting parameters for title and labels 'start.time' which sets the first time value

Details

plot.summary.tdlm

Value

A plot of distributed lag effect estimated with tdlm

plot.summary.tdlmm *Plots DLMMs for model summary of class 'tdlmm'*

Description

Method for plotting DLMMs for model summary of class 'tdlmm'. Includes plots for marginal exposure effects as well as interactions between two exposures.

Usage

```
## S3 method for class 'summary.tdlmm'
plot(
  x,
  type = "marginal",
  exposure1 = NULL,
  exposure2 = NULL,
  time1 = c(),
  time2 = c(),
  show.cw = TRUE,
  cw.plots.only = TRUE,
  trueDLM = NULL,
  scale = NULL,
  ...
)
```

Arguments

x an object of type 'summary.tdlmm' from summary.tdlmm() output
 type plot type, 'marginal' (default)
 exposure1 exposure for plotting DLM
 exposure2 exposure paired with 'exposure1' for plotting interaction
 time1 plot a cross section from an interaction plot at specific time for 'exposure1'

time2	plot a cross section from an interaction plot at specific time for 'exposure2'
show.cw	indicate location of critical windows in interaction plot with red points
cw.plots.only	show only plots with critical windows
trueDLM	A vector of true effects that can be obtained from the simulated data. Only applicable for simulation studies
scale	default = NULL, if scale is not NULL, the effects are exponentiated
...	additional plotting parameters for title and labels

Details

plot.summary.tdlnm

Value

A plot of distributed lag effect or interaction surface estimated with tdlnm

plot.summary.tdlnm *Returns variety of plots for model summary of class 'tdlnm'*

Description

Method for returning variety of plots for model summary of class 'tdlnm'

Usage

```
## S3 method for class 'summary.tdlnm'
plot(x, plot.type = "mean", val = c(), time = c(), ...)
```

Arguments

x	object of class 'summary.tdlnm', output of summary of 'tdlnm'
plot.type	string indicating plot type, options are 'mean' (default) which shows mean exposure-time response surface, 'cumulative' which shows the cumulative effects per exposure-concentration level, 'effect' which returns a grid of exposure concentration and lag to determine if the credible interval contains zero, with the direction of the effect indicated, 'se', 'ci-min', 'ci-max', 'slice' which takes a slice of the plot at a given 'val' or 'time', 'animate' which creates an animation of slices of the surface plot across exposure values (requires package gganimate)
val	exposure value for slice plot
time	time value for slice plot
...	additional plotting parameters for title and labels 'flab' which sets the effect label for surface plots, 'start.time' which sets the first time value

Details

plot.summary.tdlnm

Value

A plot of distributed lag effect estimated with tdlnm

pm25Exposures	<i>PM2.5 Exposure data</i>
---------------	----------------------------

Description

Data.frame containing a sample of weekly average PM2.5 exposures across a range of states/counties. The PM2.5 data was downloaded from US EPA (https://aqs.epa.gov/aqsweb/airdata/download_files.html) daily data summaries and averaged by week. Forty-week ranges were assess for non-missingness and grouped for this dataset.

Usage

```
data(pm25Exposures)
```

Format

data.frame; columns: S = state, C = city, 1-40 = weekly exposure data

Source

https://aqs.epa.gov/aqsweb/airdata/download_files.html

References

<https://www.epa.gov/outdoor-air-quality-data>

ppRange	<i>Makes a 'pretty' output of a group of numbers</i>
---------	--

Description

Method for making a 'pretty' output of a group of numbers. For example: 2,3,4,5,8,9,12,15,16 becomes 2-5,8-9,12,15-16

Usage

```
ppRange(r)
```

Arguments

r set of integers to make 'pretty'

Details

ppRange

Value

character string of values representing 'r'

predict	<i>predict</i>
---------	----------------

Description

predict generic function for S3method

Usage

```
predict(  
  x,  
  new.data,  
  new.exposure.data,  
  ci.level = 0.95,  
  type = "response",  
  outcome = NULL,  
  fixed.idx = list(),  
  est.dlm = FALSE,  
  verbose = TRUE,  
  ...  
)  
  
## S3 method for class 'hdlm'  
predict(  
  x,  
  new.data,  
  new.exposure.data,  
  ci.level = 0.95,  
  type = "response",  
  outcome = NULL,  
  fixed.idx = list(),  
  est.dlm = FALSE,  
  verbose = TRUE,  
  ...  
)  
  
## S3 method for class 'hdlmm'  
predict(  
  x,  
  new.data,
```

```

new.exposure.data,
ci.level = 0.95,
type = "response",
outcome = NULL,
fixed.idx = list(),
est.dlm = FALSE,
verbose = TRUE,
...
)

```

Arguments

<code>x</code>	fitted dlmtree model with class 'hdlm', 'hdlmm'
<code>new.data</code>	new data frame which contains the same covariates and modifiers used to fit the model
<code>new.exposure.data</code>	new data frame/list which contains the same length of exposure lags used to fit the model
<code>ci.level</code>	credible interval level for posterior predictive distribution
<code>type</code>	type of prediction: "response" (default) or "waic". "waic" must be specified with 'outcome' parameter
<code>outcome</code>	outcome required for WAIC calculation
<code>fixed.idx</code>	fixed index
<code>est.dlm</code>	flag for estimating dlm effect
<code>verbose</code>	TRUE (default) or FALSE: print output
<code>...</code>	not used

Value

list with the following elements:

ztg posterior predictive mean of fixed effect

ztg.lims lower/upper bound of posterior predictive distribution of fixed effect

dlmest estimated exposure effect

dlmest.lower lower bound of estimated exposure effect

dlmest.upper upper bound of estimated exposure effect

fhat posterior predictive mean of exposure effect

fhat.lims lower/upper bound of posterior predictive distribution of exposure effect

y posterior predictive mean

y.lims lower/upper bound of posterior predictive distribution

print

print

Description

print generic function for S3method

Usage

```
print(x, ...)  
  
## S3 method for class 'tdlnm'  
print(x, ...)  
  
## S3 method for class 'tdlm'  
print(x, ...)  
  
## S3 method for class 'tdlmm'  
print(x, ...)  
  
## S3 method for class 'hdlm'  
print(x, ...)  
  
## S3 method for class 'hdlmm'  
print(x, ...)  
  
## S3 method for class 'monotone'  
print(x, ...)  
  
## S3 method for class 'summary.hdlm'  
print(x, digits = 3, ...)  
  
## S3 method for class 'summary.hdlmm'  
print(x, digits = 3, ...)  
  
## S3 method for class 'summary.monotone'  
print(x, digits = 3, ...)  
  
## S3 method for class 'summary.tdlm'  
print(x, digits = 3, ...)  
  
## S3 method for class 'summary.tdlmm'  
print(x, digits = 3, ...)  
  
## S3 method for class 'summary.tdlnm'  
print(x, digits = 3, ...)
```

Arguments

x	An object of class 'tdlm', 'tdlmm', 'tdlnm', 'hdlm', 'hdlmm', 'monotone', representing a fitted model using dlmtree(); or a summary object produced by applying summary() to one of these model objects.
...	additional parameters
digits	number of decimal places to round the numeric values to

Value

For a fitted model object, prints an assorted model output including model formula call and available methods. For a summary object, prints a summary output of a model fit in the R console.

rcpp_pgdraw	<i>Multiple draw polya gamma latent variable for var c[i] with size b[i]</i>
-------------	--

Description

Multiple draw polya gamma latent variable for var c[i] with size b[i]

Usage

```
rcpp_pgdraw(b, z)
```

Arguments

b	vector of binomial sizes
z	vector of parameters

Value

Eigen::VectorXd

rtmvnorm	<i>Truncated multivariate normal sampler, mean mu, cov sigma, truncated (0, Inf)</i>
----------	--

Description

Truncated multivariate normal sampler, mean mu, cov sigma, truncated (0, Inf)

Usage

```
rtmvnorm(mu, sigma, iter)
```

Arguments

mu	vector of mean parameters
sigma	covariance matrix
iter	number of iterations

Value

VectorXd

ruleIdx	<i>Calculates the weights for each modifier rule</i>
---------	--

Description

Method for calculating the weights for each modifier rule

Usage

```
ruleIdx(mod, mem.safe = FALSE)
```

Arguments

mod	a list of modifier splitting rules
mem.safe	boolean memory parameter

Value

A list of weights per rule with modifiers

scaleModelMatrix	<i>Centers and scales a matrix</i>
------------------	------------------------------------

Description

Method for centering and scaling a matrix

Usage

```
scaleModelMatrix(M)
```

Arguments

M	a matrix to center and scale
---	------------------------------

Details

scaleModelMatrix

Value

a scaled matrix

shiny

shiny

Description

shiny generic function for S3method

Usage

```
shiny(fit)
```

```
## S3 method for class 'hdlm'
```

```
shiny(fit)
```

```
## S3 method for class 'hdlmm'
```

```
shiny(fit)
```

Arguments

`fit` object of class 'hdlm', 'hdlmm' to which S3method is applied

Value

shiny interface for further analysis on heterogeneous analyses. The interface includes tabs for modifier selection, personalized exposure effects and subgroup-specific effects.

sim.hdlmm

Creates simulated data for HDLM & HDLMM

Description

Method for creating simulated data for HDLM & HDLMM

Usage

```
sim.hdlmm(  
  sim = "A",  
  n = 1000,  
  error = 1,  
  effect.size = 1,  
  exposure.data = NULL  
)
```

Arguments

sim	character (A - E) specifying simulation scenario
n	sample size
error	positive scalar specifying error variance for Gaussian response
effect.size	the effect size of the window of susceptibility
exposure.data	exposure data. A matrix of exposure data for simulation A, B, C and a named list of exposure data for simulation D, E

Details

sim.hdlmm

Simulation scenarios:

- Scenario A: Two subgroups with early/late windows determined by continuous and binary modifiers
- Scenario B: Two subgroups with scaled effect determined by a continuous modifier
- Scenario C: No heterogeneity i.e., same effect on all individuals
- Scenario D: Three subgroups with three corresponding exposures. Subgroups are determined by continuous and binary modifiers
- Scenario E: Two subgroups with two exposures. First group is associated with the scaled main effect and lagged interaction while the second group is only associated with the scaled main effect, no interaction.

Value

Simulated data and true parameters

Examples

```
sim.hdlmm(sim = "A", n = 1000)
```

 sim.tdlmm

Creates simulated data for TDLM & TDLMM

Description

Method for creating simulated data for TDLM & TDLMM

Usage

```
sim.tdlmm(
  sim = "A",
  n = 5000,
  error = 10,
  mean.p = 0.5,
  prop.active = 0.05,
  explist = NULL,
  r = 1
)
```

Arguments

sim	character (A - F) specifying simulation scenario
n	sample size for simulation
error	positive scalar specifying error variance for Gaussian response
mean.p	scalar between zero and one specifying mean probability for simulation scenario A
prop.active	proportion of active exposures for simulation scenario C
explist	named list of exposure data
r	dispersion parameter of negative binomial distribution

Details

sim.tdlmm

Simulation scenarios:

- Scenario A: Binary response with single exposure effect
- Scenario B: Continuous response with main effect of PM2.5 and interaction
- Scenario C: Continuous response to test exposure selection using exposure
- Scenario D: Continuous response to test exposure selection using one exposure of main effect and two interaction effects among four exposures
- Scenario E: Zero-inflated count response with single exposure effect
- Scenario F: Zero-inflated count response with single exposure effect with main effect of PM2.5 and interaction

Value

Simulated data and true parameters

Examples

```
sim.tdlmm(sim = "A", mean.p = 0.5, n = 1000)
```

sim.tdlnm

Creates simulated data for TDLNM

Description

Method for creating simulated data for TDLNM

Usage

```
sim.tdlnm(sim = "A", error.to.signal = 1)
```

Arguments

sim character (A - D) specifying simulation scenario
error.to.signal scalar value setting error: $\sigma^2/\text{var}(f)$

Details

sim.tdlnm

Simulation scenarios:

- Scenario A: Piecewise constant effect
- Scenario B: Linear effect
- Scenario C: Logistic effect, piecewise in time
- Scenario D: Logistic effect, smooth in time

Value

Simulated data and true parameters

Examples

```
sim.tdlnm(sim = "A", error.to.signal = 1)
```

splitPIP	<i>Calculates the posterior inclusion probability (PIP).</i>
----------	--

Description

Calculates the posterior inclusion probability (PIP).

Usage

```
splitPIP(dlnm, nlags, niter)
```

Arguments

dlnm	A numeric matrix containing the model fit information
nlags	total number of lags
niter	number of mcmc iterations

Value

A matrix of split counts per mcmc

splitpoints	<i>Determines split points for continuous modifiers</i>
-------------	---

Description

Method for determining split points for continuous modifiers

Usage

```
splitpoints(object, var, round = NULL)
```

Arguments

object	An object of class 'hdlm', 'hdlmm'
var	The name of a continuous variable for which the split points will be reported
round	The number of decimal places to round the variable (var) to. No rounding occurs if round=NULL (default) For positive integer values of round, the variable will be rounded and split points will be reported at the resulting level

Details

splitpoints

Value

A data frame with split points and the probability that a split point was \geq that split point value

Examples

```
# Split points with HDLM
D <- sim.hdlmm(sim = "B", n = 1000)
fit <- dlmtree(y ~ .,
               data = D$dat,
               exposure.data = D$exposures,
               dlm.type = "linear",
               family = "gaussian",
               het = TRUE)
splitpoints(fit, var = "mod_num", round = 2)
splitpoints(fit, var = "mod_scale", round = 2)
```

summary

summary

Description

summary generic function for S3method

Usage

```
summary(x, conf.level = 0.95, ...)

## S3 method for class 'hdlm'
summary(x, conf.level = 0.95, mcmc = FALSE, ...)

## S3 method for class 'hdlmm'
summary(x, conf.level = 0.95, mcmc = FALSE, ...)

## S3 method for class 'monotone'
summary(
  x,
  conf.level = 0.95,
  pred.at = NULL,
  cenva1 = 0,
  exposure.se = NULL,
  mcmc = FALSE,
  verbose = TRUE,
  ...
)

## S3 method for class 'tdlm'
```

```
summary(x, conf.level = 0.95, mcmc = FALSE, ...)
```

```
## S3 method for class 'tdlmm'
```

```
summary(
  x,
  conf.level = 0.95,
  marginalize = "mean",
  log10BF.crit = 0.5,
  mcmc = FALSE,
  verbose = TRUE,
  ...
)
```

```
## S3 method for class 'tdlnm'
```

```
summary(
  x,
  conf.level = 0.95,
  pred.at = NULL,
  cenval = 0,
  exposure.se = NULL,
  mcmc = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

<code>x</code>	an object of class 'tdlm', 'tdlmm', 'tdlnm', 'hdlm', 'hdlmm', 'monotone'
<code>conf.level</code>	confidence level for computation of credible intervals
<code>...</code>	additional parameters
<code>mcmc</code>	keep all mcmc iterations (large memory requirement)
<code>pred.at</code>	numerical vector of exposure values to make predictions for at each time period
<code>cenval</code>	scalar exposure value that acts as a reference point for predictions at all other exposure values
<code>exposure.se</code>	scalar smoothing factor, if different from model
<code>verbose</code>	show progress in console
<code>marginalize</code>	value(s) for calculating marginal DLMS, defaults to "mean", can also specify a percentile from 1-99 for all other exposures, or a named vector with specific values for each exposure
<code>log10BF.crit</code>	Bayes Factor criteria for selecting exposures and interactions, such that $\log_{10}(\text{BayesFactor}) > x$. Default = 0.5.

Value

list of summary outputs of the model fit

tdlmm_Cpp	<i>dlmtree model with tdlmm approach</i>
-----------	--

Description

dlmtree model with tdlmm approach

Usage

```
tdlmm_Cpp(model)
```

Arguments

model A list of parameter and data contained for the model fitting

Value

A list of dlmtree model fit, mainly posterior mcmc samples

tdlnm_Cpp	<i>dlmtree model with tdlnm approach</i>
-----------	--

Description

dlmtree model with tdlnm approach

Usage

```
tdlnm_Cpp(model)
```

Arguments

model A list of parameter and data contained for the model fitting

Value

A list of dlmtree model fit, mainly posterior mcmc samples

zeroToInfNormCDF	<i>Integrates (0,inf) over multivariate normal</i>
------------------	--

Description

Integrates (0,inf) over multivariate normal

Usage

```
zeroToInfNormCDF(mu, sigma)
```

Arguments

mu	vector of mean parameters
sigma	covariance matrix

Value

double

zinbCo	<i>Time-series exposure data for ZINB simulated data</i>
--------	--

Description

Data.frame containing a sample of weekly average wildfire PM, PM2.5, O3 across a range of counties of Colorado. The exposure data was downloaded from US EPA (https://aqs.epa.gov/aqsweb/airdata/download_files.html) daily data summaries and averaged by week.

Usage

```
data(zinbCo)
```

Format

data.frame;

Source

https://aqs.epa.gov/aqsweb/airdata/download_files.html

References

<https://www.epa.gov/outdoor-air-quality-data>

Index

* datasets

- coExp, 4
- exposureCov, 23
- pm25Exposures, 30
- zinbCo, 44

adj_coexposure, 3

coExp, 4
combine.models, 5
combine.models.tdlmm, 5
cppIntersection, 6

diagnose (diagnose.summary.hdlm), 6
diagnose.summary.hdlm, 6
dlmEst, 7
dlmtree, 8
dlmtree.control.diagnose, 9, 11
dlmtree.control.family, 9, 12
dlmtree.control.het, 9, 13
dlmtree.control.hyper, 9, 14
dlmtree.control.mcmc, 9, 14
dlmtree.control.mix, 9, 15
dlmtree.control.monotone, 9, 15
dlmtree.control.tdlnm, 9, 16
dlmtreeGPFixedGaussian, 17
dlmtreeGPGaussian, 17
dlmtreeHDLMGaussian, 18
dlmtreeHDLMMGaussian, 18
dlmtreeTDLM_cpp, 20
dlmtreeTDLMFixedGaussian, 19
dlmtreeTDLMNestedGaussian, 19
dlnmEst, 20
dlnmPEst, 21
drawTree, 21

estDLM, 22
exposureCov, 23

get_sbd_dlmtree, 23

mixEst, 25
monotdlm_Cpp, 25

pip, 26
plot.summary.monotone, 27
plot.summary.tdlm, 27
plot.summary.tdlmm, 28
plot.summary.tdlnm, 29
pm25Exposures, 30
ppRange, 30
predict, 31
print, 33

rcpp_pgdraw, 34
rtmvnorm, 34
ruleIdx, 35

scaleModelMatrix, 35
shiny, 36
sim.hdlmm, 36
sim.tdlmm, 38
sim.tdlnm, 39
splitPIP, 40
splitpoints, 40
summary, 41

tdlmm_Cpp, 43
tdlnm_Cpp, 43

zeroToInfNormCDF, 44
zinbCo, 44