Package ‘nlmixr2extra’

January 31, 2024

Title  Nonlinear Mixed Effects Models in Population PK/PD, Extra Support Functions

Version  2.0.9

Maintainer  Matthew Fidler <matthew.fidler@gmail.com>


Depends  R (>= 4.0)

License  GPL (>= 3)


BugReports  https://github.com/nlmixr2/nlmixr2extra/issues/

Imports  checkmate, cli (>= 3.4.0), crayon, data.table, digest, ggplot2, ggtext, knitr, lotri, methods, nlme, nlmixr2est (>= 2.1.1), Rcpp, rxode2 (>= 2.0.10), stats, symengine, utils

Suggests  brms, nlmixr2data, testthat (>= 3.0.0), withr, dplyr, devtools, rxode2parse

LinkingTo  Rcpp, RcppArmadillo

Biarch  true

Config/testthat/edition  3

Encoding  UTF-8

NeedsCompilation  yes

RoxygenNote  7.3.1

LazyData  true
adaptivelassoCoefficients

Return Adaptive lasso coefficients after finding optimal t

Description

Return Adaptive lasso coefficients after finding optimal t
adaptivelassoCoefficients

Usage

adaptivelassoCoefficients(
  fit,
  varsVec,
  covarsVec,
  catvarsVec,
  constraint = 1e-08,
  stratVar = NULL,
  ...
)

Arguments

fit nlmixr2 fit.
varsVec character vector of variables that need to be added
covarsVec character vector of covariates that need to be added
catvarsVec character vector of categorical covariates that need to be added
constraint theta cutoff. below cutoff then the theta will be fixed to zero.
stratVar A variable to stratify on for cross-validation.
... Other parameters to be passed to optmalTvaluelasso

Value

return data frame of final lasso coefficients

Author(s)

Vishal Sarsani

Examples

## Not run:
one.cmt <- function() {
  ini({
    tka <- 0.45; label("Ka")
    tcl <- log(c(0, 2.7, 100)); label("Cl")
    tv <- 3.45; label("V")
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
  })
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    linCmt() ~ add(add.sd)
  })
addCatCovariates

Make dummy variable cols and updated covarsVec

Description
Make dummy variable cols and updated covarsVec

Usage
addCatCovariates(data, covarsVec, catcovarsVec)

Arguments
  data            data frame used in the analysis
  covarsVec       character vector of covariates that need to be added
  catcovarsVec    character vector of categorical covariates that need to be added

Value
  return updated Data along with the updated covarsVec

Author(s)
  Vishal Sarsani
addorremoveCovariate

Description
Add covariate

Usage
addorremoveCovariate(ui, varName, covariate, add = TRUE)

Arguments
- ui: compiled rxode2 nlmir2 model or fit
- varName: the variable name to which the given covariate is to be added
- covariate: the covariate that needs string to be constructed
- add: boolean indicating if the covariate needs to be added or removed.

Author(s)
Matthew Fidler, Vishal Sarsani

adjustedlassoCoefficients

Description
Return Adjusted adaptive lasso coefficients after finding optimal t

Usage
adjustedlassoCoefficients(
  fit,
  varsVec,
  covarsVec,
  catvarsVec,
  constraint = 1e-08,
  stratVar = NULL,
  ...
)
Arguments

fit nlmixr2 fit.
varsVec character vector of variables that need to be added
covarsVec character vector of covariates that need to be added
catvarsVec character vector of categorical covariates that need to be added
constraint theta cutoff. below cutoff then the theta will be fixed to zero.
stratVar A variable to stratify on for cross-validation.
... Other parameters to be passed to optimalTvaluelasso

Value

return data frame of final lasso coefficients

Author(s)

Vishal Sarsani

Examples

## Not run:
one.cmt <- function() {
ini({
tka <- 0.45; label("Ka")
tcl <- log(c(0, 2.7, 100)); label("Cl")
tv <- 3.45; label("V")
etak <- 0.6
etacl <- 0.3
etav <- 0.1
add.sd <- 0.7
})
model({
ka <- exp(tka + etak)
cl <- exp(tcl + etacl)
v <- exp(tv + etav)
linCmt() ~ add(add.sd)
})
}
d <- nlmixr2data::theo_sd
d$SEX <- 0
d$SEX[d$ID<=6] <- 1
fit <- nlmixr2(one.cmt, d, est = "saem", control = list(print = 0))
varsVec <- c("ka","cl","v")
covarsVec <- c("WT")
catvarsVec <- c("SEX")

# Adaptive Lasso coefficients:
**backwardSearch**

lassoDf <- adjustedlassoCoefficients(fit,varsVec,covarsVec,catvarsVec)

## End(Not run)

### Description
Backward covariate search

### Usage
backwardSearch(
  varsVec,
  covarsVec,
  catvarsVec = NULL,
  fitorig,
  fitupdated,
  pVal = 0.01,
  reFitCovars = FALSE,
  outputDir,
  restart = FALSE
)

### Arguments
- **varsVec** character vector of variables that need to be added
- **covarsVec** character vector of covariates that need to be added
- **catvarsVec** character vector of categorical covariates that need to be added
- **fitorig** the original 'fit' object before forward search
- **fitupdated** the updated 'fit' object, if any, after the forward search
- **pVal** p-value that should be used for selecting covariates in the forward search
- **reFitCovars** if the covariates should be added before performing backward search - useful for directly performing backward search without forward search; default is FALSE
- **outputDir** the name of the output directory that stores the covariate search result
- **restart** a boolean that controls if the search should be restarted; default is FALSE

### Value
returns the updated 'fit' object at the end of the backward search and a table of information for all the covariates tested

### Author(s)
Vipul Mann, Matthew Fidler, Vishal Sarsani
bootstrapFit

**Description**

Bootstrap input dataset and rerun the model to get confidence bounds and aggregated parameters

---

bootplot

*Produce delta objective function for bootstrap*

**Description**

Produce delta objective function for bootstrap

**Usage**

```r
bootplot(x, ...)  
```

```r
## S3 method for class 'nlmixr2FitCore'
bootplot(x, ...)
```

**Arguments**

- `x` fit object
- `...` other parameters

**Value**

Fit traceplot or nothing.

**Author(s)**

Vipul Mann, Matthew L. Fidler

**References**

R Niebecker, MO Karlsson. (2013) *Are datasets for NLME models large enough for a bootstrap to provide reliable parameter uncertainty distributions?* PAGE 2013. [https://www.page-meeting.org/?abstract=2899](https://www.page-meeting.org/?abstract=2899)
Usage

bootstrapFit(
  fit,
  nboot = 200,
  nSampIndiv,
  stratVar,
  stdErrType = c("perc", "se"),
  ci = 0.95,
  pvalues = NULL,
  restart = FALSE,
  plotHist = FALSE,
  fitName = as.character(substitute(fit))
)

Arguments

fit          the nlmixr2 fit object
nboot        an integer giving the number of bootstrapped models to be fit; default value is 200
nSampIndiv   an integer specifying the number of samples in each bootstrapped sample; default is the number of unique subjects in the original dataset
stratVar     Variable in the original dataset to stratify on; This is useful to distinguish between sparse and full sampling and other features you may wish to keep distinct in your bootstrap
stdErrType   This gives the standard error type for the updated standard errors; The current possibilities are: "perc" which gives the standard errors by percentiles (default) or "se" which gives the standard errors by the traditional formula.
ci           Confidence interval level to calculate. Default is 0.95 for a 95 percent confidence interval
pvalues      a vector of pvalues indicating the probability of each subject to get selected; default value is NULL implying that probability of each subject is the same
restart      A boolean to try to restart an interrupted or incomplete boostrap. By default this is FALSE
plotHist     A boolean indicating if a histogram plot to assess how well the bootstrap is doing. By default this is turned off (FALSE)
fitName      is the fit name that is used for the name of the boostrap files. By default it is the fit provided though it could be something else.

Value

Nothing, called for the side effects; The original fit is updated with the bootstrap confidence bands

Author(s)

Vipul Mann, Matthew Fidler
Examples

```r
## Not run:
one.cmt <- function() {
  ini({
    tka <- 0.45; label("Ka")
    tcl <- 1; label("Cl")
    tv <- 3.45; label("V")
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
  })
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    linCmt() ~ add(add.sd)
  })
}
fit <- nlmixr2(one.cmt, nlmixr2data::theo_sd, est = "saem", control = list(print = 0))

withr::with_tempdir({ # Run example in temp dir
  bootstrapFit(fit, nboot = 5, restart = TRUE) # overwrites any of the existing data or model files
  bootstrapFit(fit, nboot = 7) # resumes fitting using the stored data and model files
  # Note this resumes because the total number of bootstrap samples is not 10
  bootstrapFit(fit, nboot=10)
  # Note the bootstrap standard error and variance/covariance matrix is retained.
  # If you wish to switch back you can change the covariance matrix by
  nlmixr2est::setCov(fit, "linFim")
  # And change it back again
  nlmixr2est::setCov(fit, "boot10")
  # This change will affect any simulations with uncertainty in their parameters
  # You may also do a chi-square diagnostic plot check for the bootstrap with
  bootplot(fit)
})

## End(Not run)
```

buildcovInfo

`Build covInfo list from varsVec and covarsVec`
buildcovInfo

Description
Build covInfo list from varsVec and covarsVec

Usage
buildcovInfo(varsVec, covarsVec)

Arguments
- varsVec: character vector of variables that need to be added
- covarsVec: character vector of covariates that need to be added

Value
covInfo list of covariate info

Author(s)
Vishal Sarsani

buildupatedUI
Build updated from the covariate and variable vector list

Description
Build updated from the covariate and variable vector list

Usage
buildupatedUI(ui, varsVec, covarsVec, add = TRUE, indep = FALSE)

Arguments
- ui: compiled rxode2 nlmir2 model or fit
- varsVec: character vector of variables that need to be added
- covarsVec: character vector of covariates that need to be added
- add: boolean indicating if the covariate needs to be added or removed
- indep: a boolean indicating if the covariates should be added independently, or sequentially (append to the previous model). only applicable to adding covariate

Value
updated ui with added covariates

Author(s)
Vishal Sarsani
covarSearchAuto  

Stepwise Covariate Model-selection (SCM) method

Description

Stepwise Covariate Model-selection (SCM) method

Usage

covarSearchAuto(
    fit, varsVec, covarsVec,
    pVal = list(fwd = 0.05, bck = 0.01),
    catvarsVec = NULL,
    searchType = c("scm", "forward", "backward"),
    restart = FALSE
)

Arguments

fit  an nlmixr2 'fit' object
varsVec a list of candidate variables to which the covariates could be added
covarsVec a list of candidate covariates that need to be tested
pVal a named list with names 'fwd' and 'bck' for specifying the p-values for the forward and backward searches, respectively
catvarsVec character vector of categorical covariates that need to be added
searchType one of 'scm', 'forward' and 'backward' to specify the covariate search method; default is 'scm'
restart a boolean that controls if the search should be restarted; default is FALSE

Value

A list summarizing the covariate selection steps and output; This list has the "summaryTable" for the overall summary of the covariate selection as well as "resFwd" for the forward selection method and "resBck" for the backward selection method.

Author(s)

Vipul Mann, Matthew Fidler, Vishal Sarsani
Examples

```r
## Not run:
one.cmt <- function() {
  ini({
    tka <- 0.45; label("Ka")
    tcl <- log(c(0, 2.7, 100)); label("Cl")
    tv <- 3.45; label("V")
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
  })
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    linCmt() ~ add(add.sd)
  })
}

fit <- nlmixr2(one.cmt, nlmixr2data::theo_sd, est = "saem", control = list(print = 0))

# with temporary directory
auto1 <- covarSearchAuto(fit, varsVec = c("ka", "cl"),
                          covarsVec = c("WT"))
```

```
## Note that this didn't include sex, add it to dataset and restart model

d <- nlmixr2data::theo_sd
d$SEX <- 0
d$SEX[d$ID<=6] <- 1

# This would restart if for some reason the search crashed:

auto2 <- covarSearchAuto(fit, varsVec = c("ka", "cl"),
                          covarsVec = c("WT"),
                          restart = TRUE)

auto3 <- covarSearchAuto(fit, varsVec = c("ka", "cl"),
                          covarsVec = c("WT"),
                          restart = TRUE,
                          searchType = "forward")
```

```r
## End(Not run)
```
foldgen

Stratified cross-validation fold generator function, inspired from the caret

Description

Stratified cross-validation fold generator function, inspired from the caret

Usage

foldgen(data, nfold = 5, stratVar = NULL)

Arguments

data  data frame used in the analysis
nfold  number of k-fold cross validations. Default is 5
stratVar  Stratification Variable. Default is NULL and ID is used for CV

Value

return data.frame with the fold column attached

Author(s)

Vishal Sarsani, caret

Examples

d <- nlmixr2data::theo_sd
d$SEX <-0
d$SEX[d$ID<=6] <-1
covarsVec <- c("WT")

# Stratified cross-validation data with CMT
df1 <- foldgen(d, nfold=5, stratVar="CMT")

# Stratified cross-validation data with ID (individual)
df2 <- foldgen(d, nfold=5, stratVar=NULL)
**forwardSearch**

**Forward covariate search**

**Description**

Forward covariate search

**Usage**

```r
forwardSearch(
  varsVec,
  covarsVec,
  catvarsVec = NULL,
  fit,
  pVal = 0.05,
  outputDir,
  restart = FALSE
)
```

**Arguments**

- `varsVec` character vector of variables that need to be added
- `covarsVec` character vector of covariates that need to be added
- `catvarsVec` character vector of categorical covariates that need to be added
- `fit` an nlmixr2 'fit' object
- `pVal` p-value that should be used for selecting covariates in the forward search
- `outputDir` the name of the output directory that stores the covariate search result
- `restart` a boolean that controls if the search should be restarted; default is FALSE

**Value**

returns the updated ‘fit’ object at the end of the forward search and a table of information for all the covariates tested

**Author(s)**

Vipul Mann, Matthew Fidler, Vishal Sarsani
horseshoeSummardf  

Create Horseshoe summary posterior estimates

Description

Create Horseshoe summary posterior estimates

Usage

horseshoeSummardf(fit, covarsVec, ...)

Arguments

- **fit**: compiled rxode2 nlmixr2 model fit
- **covarsVec**: character vector of covariates that need to be added
- **...**: other parameters passed to brm(): warmup = 1000, iter = 2000, chains = 4, cores = 4, control = list(adapt_delta = 0.99, max_treedepth = 15)

Value

Horse shoe Summary data frame of all covariates

Author(s)

Vishal Sarsani, Christian Bartels

Examples

```r
## Not run:
one.cmt <- function() {
  ini({
    tka <- 0.45; label("Ka")
tcl <- log(c(0, 2.7, 100)); label("Cl")
tv <- 3.45; label("V")
et.ka ~ 0.6
eta.cl ~ 0.3
ta.v ~ 0.1
add.sd <- 0.7
})
model({
  ka <- exp(tka + eta.ka)
cl <- exp(tcl + eta.cl)
v <- exp(tv + eta.v)
linCmt() ~ add(add.sd)
})
}
d <- nlmixr2data::theo_sd
fit <- nlmixr2(one.cmt, d, est = "saem", control = list(print = 0))
```

covarsVec <- c("WT")

# Horseshoe summary posterior estimates:
#hsDf <- horseshoeSummardf(fit,covarsVec,cores=2)
#brms sometimes may throw a Error in sink(type = "output")
#Issue Should be fixed by uninstalling and re-installing rstan
## End(Not run)

knit_print.nlmixr2FitCore

---

**knit_print.nlmixr2FitCore**

Extract the equations from an nlmixr2/rxode2 model to produce a ‘LaTeX’ equation.

---

**Description**

Extract the equations from an nlmixr2/rxode2 model to produce a ‘LaTeX’ equation.

**Usage**

```r
## S3 method for class 'nlmixr2FitCore'
knit_print(x, ..., output = "equations")

## S3 method for class 'rxUi'
knit_print(x, ...)
```

**Arguments**

- `x`: The model to extract equations from
- `...`: Ignored
- `output`: The type of output to request (currently, just "equations")

**lassoCoefficients**

Return Final lasso coefficients after finding optimal t

---

**Description**

Return Final lasso coefficients after finding optimal t
Usage

lassoCoefficients(
    fit,
    varsVec,
    covarsVec,
    catvarsVec,
    constraint = 1e-08,
    stratVar = NULL,
    ...
)

Arguments

fit nlmixr2 fit.
varsVec character vector of variables that need to be added
covarsVec character vector of covariates that need to be added
catvarsVec character vector of categorical covariates that need to be added
constraint theta cutoff. below cutoff then the theta will be fixed to zero
stratVar A variable to stratify on for cross-validation
...
Other parameters to be passed to optimalTvaluelasso

Value

return data frame of final lasso coefficients

Author(s)

Vishal Sarsani

Examples

## Not run:
one.cmt <- function() {
in{
    tka <- 0.45; label("Ka")
    tcl <- log(c(0, 2.7, 100)); label("Cl")
    tv <- 3.45; label("V")
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
}
model{
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    linCmt() ~ add(add.sd)
}
```r
lassoSummardf

} d <- nlmixr2data::theo_sd
d$SEX <- 0
d$SEX[d$ID<=6] <- 1

fit <- nlmixr2(one.cmt, d, est = "saem", control = list(print = 0))
varsVec <- c("ka","cl","v")
covarsVec <- c("WT")
catvarsVec <- c("SEX")

# Lasso coefficients:
lassoDf <- lassoCoefficients(fit, varsVec, covarsVec, catvarsVec, constraint=1e-08, stratVar = NULL)

# End(Not run)

lassoSummardf Create Lasso summary posterior estimates

Description

Create Lasso summary posterior estimates

Usage

lassoSummardf(fit, covarsVec, ...)

Arguments

fit compiled rxode2 nlmixr2 model fit
covarsVec character vector of covariates that need to be added
... other parameters passed to brm(): warmup = 1000, iter = 2000, chains = 4, cores = 4, control = list(adapt_delta = 0.99, max_treedepth = 15)

Value

Horse shoe Summary data frame of all covariates

Author(s)

Vishal Sarsani, Christian Bartels
normalizedData

Function to return data of normalized covariates

Description

Function to return data of normalized covariates

Usage

normalizedData(data, covarsVec, replace = TRUE)

Arguments

data a dataframe with covariates to normalize
covarsVec a list of covariate names (parameters) that need to be estimates
replace replace the original covariate data with normalized data for easier updated model.
**Value**

data frame with all normalized covariates

**Author(s)**

Vishal Sarsani

**Examples**

d <- nlmixr2data::theo_sd
d$SEX <-0
d$SEX[d$ID<=6] <-1
covarsVec <- c("WT")

# Normalized covariate (replaced)
df1 <- normalizedData(d,covarsVec,replace=TRUE)

# Normalized covariate (without replacement)
df2 <- normalizedData(d,covarsVec,replace=FALSE)

---

**optimUnisampling**

Sample from uniform distribution by optim

**Description**

Sample from uniform distribution by optim

**Usage**

optimUnisampling(xvec, N = 1000, medValue, floorT = TRUE)

**Arguments**

- **xvec**
  A vector of min,max values. Ex:c(10,20)
- **N**
  Desired number of values
- **medValue**
  Desired Median
- **floorT**
  boolean indicating whether to round up

**Value**

Samples with approx desired median.

**Author(s)**

Vishal Sarsani
Examples

```r
# Simulate 1000 creatine clearance values with median of 71.7 within range of c(6.7,140)
creatCl <- optimUnisampling(xvec=c(6.7,140), N=1000, medValue = 71.7, floorT=FALSE)
```

### preconditionFit

**Linearily re-parameterize the model to be less sensitive to rounding errors**

**Description**

Linearily re-parameterize the model to be less sensitive to rounding errors

**Usage**

```r
preconditionFit(fit, estType = c("full", "posthoc", "none"), ntry = 10L)
```

**Arguments**

- **fit**: A nlmixr2 fit to be preconditioned
- **estType**: Once the fit has been linearly reparameterized, should a "full" estimation, "posthoc" estimation or simply a estimation of the covariance matrix "none" before the fit is updated
- **ntry**: number of tries before giving up on a pre-conditioned covariance estimate

**Value**

A nlmixr2 fit object that was preconditioned to stabilize the variance/covariance calculation

**References**


### regularmodel

**Regular lasso model**

**Description**

Regular lasso model
Usage

regularmodel(
  fit,
  varsVec,
  covarsVec,
  catvarsVec,
  constraint = 1e-08,
  lassotype = c("regular", "adaptive", "adjusted"),
  stratVar = NULL,
  ...
)

Arguments

fit nlmixr2 fit.
varsVec character vector of variables that need to be added
covarsVec character vector of covariates that need to be added
catvarsVec character vector of categorical covariates that need to be added
constraint theta cutoff. below cutoff then the theta will be fixed to zero.
lassotype must be 'regular', 'adaptive', 'adjusted'
stratVar A variable to stratify on for cross-validation.
... Other parameters to be passed to optimalTvaluelasso

Value

return fit of the selected lasso coefficients

Author(s)

Vishal Sarsani

Examples

## Not run:
one.cmt <- function() {
  ini(
    tka <- 0.45; label("Ka")
    tcl <- log(c(0, 2.7, 100)); label("Cl")
    tv <- 3.45; label("V")
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
  )
  model(
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
  )
}
```r
linCmt() ~ add(add.sd)
}
}
d <- nlmixr2data::theo_sd
d$SEX <- 0
d$SEX[d$ID<=6] <- 1
fit <- nlmixr2(one.cmt, d, est = "saem", control = list(print = 0))
varsVec <- c("ka","cl","v")
covarsVec <- c("WT")
catvarsVec <- c("SEX")

# Model fit with regular lasso coefficients:
lassoDf <- regularmodel(fit,varsVec,covarsVec,catvarsVec)
# Model fit with adaptive lasso coefficients:
lassoDf <- regularmodel(fit,varsVec,covarsVec,catvarsVec,lassotype='adaptive')
# Model fit with adaptive-adjusted lasso coefficients:
lassoDf <- regularmodel(fit,varsVec,covarsVec,catvarsVec, lassotype='adjusted')
## End(Not run)
```

---

**Example single dose Theophylline ODE model**

**Description**

This is a nlmixr2 model that is pre-run so that it can be used in package testing and development. It is regenerated whenever binaries of nlmixr2extra are created. If there is a binary incompatibility between the fit objects, a simple rerun of the installation will fix this nlmixr2 fit object.

**Format**

A (modified) data frame with 132 rows and 22 columns.

<table>
<thead>
<tr>
<th>ID</th>
<th>Patient identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>Time (hr)</td>
</tr>
<tr>
<td>DV</td>
<td>Dependent variable (concentration)</td>
</tr>
<tr>
<td>PRED</td>
<td>Predictions without any between subject variability</td>
</tr>
<tr>
<td>RES</td>
<td>Population Residual</td>
</tr>
<tr>
<td>WRES</td>
<td>Weighted Residuals under the FO assumption</td>
</tr>
<tr>
<td>IPRED</td>
<td>Individual Predictions</td>
</tr>
<tr>
<td>IRES</td>
<td>Individual Residuals</td>
</tr>
<tr>
<td>IWRES</td>
<td>Individual Weighted Residuals</td>
</tr>
</tbody>
</table>
CPRED  Conditional Prediction under the FOCE assumption
CRES  Conditional Residuals under the FOCE assumption
CWRES  Conditional Weighted Residuals under the FOCE assumption
eta.ka  Between subject changes for ka
eta.cl  Between subject changes for v
depot  amount in the depot compartment
center  amount in the central compartment
ka  Individual ka values
cl  Individual cl values
v  Individual volume of distribution
tad  Time after dose
dosenum  Dose number
Index

adaptivelassocoeficients, 2
addcatcovariates, 4
addorremovecovariate, 5
adjustedlassocoeficients, 5
backwardsearch, 7
bootplot, 8
bootstrapfit, 8
buildcovinfo, 10
buildupdatedui, 11
covarsearchauto, 12
foldgen, 14
forwardsearch, 15
horseshoessummardf, 16
knit_print.nlmixr2fitcore, 17
knit_print.rxui
  (knit_print.nlmixr2fitcore), 17
lassocoeficients, 17
lassosummardf, 19
normalizeddata, 20
optimunisampling, 21
preconditionfit, 22
regularmodel, 22
thoefitode, 24