Package ‘npde’

February 20, 2015

Type Package
Title Normalised prediction distribution errors for nonlinear mixed-effect models
Version 2.0
Date 2012-08-15
Author Emmanuelle Comets, Karl Brendel, Thi Huyen Tram Nguyen, France Mentre.
Maintainer Emmanuelle Comets <emmanuelle.comets@inserm.fr>
Description Routines to compute normalised prediction distribution errors, a metric designed to evaluate non-linear mixed effect models such as those used in pharmacokinetics and pharmacodynamics
License GPL (>= 2)
LazyLoad yes
LazyData yes
Depends methods, mclust
Imports graphics, stats
Collate 'global.R' 'NpdeData.R' 'NpdeRes.R' 'NpdeObject.R'
'func_methods.R' 'func_plots.R' 'main.R' 'zzz.R'
Repository CRAN
Date/Publication 2012-10-15 15:31:07
NeedsCompilation no

R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>npde-package</td>
<td>2</td>
</tr>
<tr>
<td>dist.pred.sim</td>
<td>3</td>
</tr>
<tr>
<td>gof.test</td>
<td>4</td>
</tr>
<tr>
<td>kurtosis</td>
<td>6</td>
</tr>
<tr>
<td>npde</td>
<td>7</td>
</tr>
<tr>
<td>npde.cens.method</td>
<td>9</td>
</tr>
</tbody>
</table>
npde-package

Normalised prediction distribution errors for nonlinear mixed-effect models

Description

Routines to compute normalised prediction distribution errors, a metric designed to evaluate nonlinear mixed effect models such as those used in pharmacokinetics and pharmacodynamics

Details

Package: npde
Type: Package
Version: 2.0
Date: 2012-08-15
License: GPL version 2 or later

See the documentation for npde or autonpde for details. A comprehensive user manual is provided in the inst directory of the package, along with a document illustrating the different graphs and graphical options. Please refer to these two guides for details, and send all comments and bug reports to Emmanuelle Comets (<emmanuelle.comets@bichat.inserm.fr>).

Author(s)

Emmanuelle Comets, Karl Brendel and France Mentre
Maintainer: Emmanuelle Comets <emmanuelle.comets@bichat.inserm.fr>
References


Examples

data(theopp)
data(simtheopp)

# Calling autonpde with dataframes

# x<-autonpde(theopp,simtheopp,1,3,4,boolsave=FALSE)
# x$npde

---

dist.pred.sim  

*Compute distribution of pd/npde using simulations*

Description

This function is used to built the distribution of pd/npde using the simulations under the model. The default is to build only the distribution of pd, and to sample from N(0,1) when building the distribution of npde under the null hypothesis.

Usage

```
dist.pred.sim(npdeObject, nsamp, ...)
```

Arguments

- **npdeObject**: an object returned by a call to `npde` or `autonpde`
- **nsamp**: number of datasets (defaults to 100 or to the number of replications if it is smaller)
- **...**: additional arguments. Currently only the value of calc.pd and calc.npde may be passed on, and will override their corresponding value in the "options" slot of npdeObject
Value

an object of class NpdeObject; the ["results"] slot will contain pd and/or npde for a sample of the simulated datasets (depending on whether calc.pd/calc.npde are ), stored in pd.sim and/or npde.sim

Author(s)

Emmanuelle Comets <emmanuelle.comets@bichat.inserm.fr>

References


See Also

npde, autonpde

Examples

data(theopp)
data(simtheopp)
x<-autonpde(theopp,simtheopp,1,3,4,bs=F)
  # Use random samples from N(0,1) to obtain a prediction interval on the empirical cdf of the npde
  plot(x,plot.type="ecdf",bands=TRUE,approx.pi=TRUE)
  # defaults to computing the pd and npde for 100 simulated datasets (in the theophylline example, this uses all the
  x<-distNPpredNPsim(x)
  # Use the npde from the simulated datasets to obtain a prediction interval on the empirical cdf
  plot(x,plot.type="ecdf",bands=TRUE,approx.pi=F)

gof.test

Description

Performs a global test on npde (default) or pd

Usage

gof.test.NpdeObject(object, which = "npde",
  parametric = TRUE, ...)

Test on npde or pd
Arguments

- **object**: an object (currently has methods for types numeric, NpdeRes and NpdeObject)
- **which**: whether the tests should be performed for npde (default), pd or npd (normalised pd)
- **parametric**: whether parametric or non-parametric tests should be applied
- **...**: additional arguments passed on to the function; special arguments are `na.action`, which controls how to handle NAs in the results, `verbose` (if FALSE, suppresses printing of the results) and `covsplit` which requests the tests to be performed split by categories or quantiles of the data. If `covsplit` is TRUE, continuous covariates will be split in 3 categories (<Q1, Q1-Q3, >Q3) (see details in the PDF documentation), but this behaviour can be overridden by passing the argument `ncat=XXX` where XXX is the number of categories to divide the continuous covariates in.

Details

If object is an NpdeObject and an argument covsplit=TRUE is given in ..., in addition to the global descriptive statistics and tests, tests will be performed for each covariate in which.cov. This argument can be set in ...; barring an explicit specification, the component which.cov of the prefs slot for a NpdeObject object will be used. The default value is which.cov="all", which produces tests for each covariate in the dataset. Two additional dataframes will then be present:

- **cov.stat**: descriptive statistics and test p-values split by covariate and by categories
- **cov.p.value**: p-values split by covariate; for each covariate, two tests are performed: the first test is a correlation test for continuous covariates and a Chi-square test for categorical covariates; the second test is defined using the p-values of the global tests split by each category, and applying a Bonferroni correction to obtain an overall p-value (see PDF documentation for details)

The p.value elements is a named vector with four components:

- **p.mean**: p-value for the mean test (Wilcoxon test if parametric=FALSE, Student test if parametric=TRUE)
- **p.var**: p-value for the variance test (parametric=FALSE, Fisher test if parametric=TRUE)
- **p.dist**: p-value for the distribution test (XXX if parametric=FALSE, XXX if parametric=TRUE)
- **p.global**: p-value for the global test (combination of the mean, variance and distribution tests with a Bonferroni correction)

Value

A list with the following elements:

- **mean**: mean
- **se.mean**: standard error of the mean
- **var**: variance
- **se.var**: standard error on variance
- **kurtosis**: kurtosis (see kurtosis)
- **skewness**: skewness (see skewness)
- **p.value**: p-values for several tests (see below)
References


See Also

*kurtosis*, *skewness*

Examples

```r
data(theopp)
```

<table>
<thead>
<tr>
<th>kurtosis</th>
<th>Kurtosis</th>
</tr>
</thead>
</table>

Description

Computes the kurtosis.

Usage

```r
kurtosis(x)
```

Arguments

- `x` a numeric vector containing the values whose kurtosis is to be computed. NA values are removed in the computation.

Details

If \( N = \text{length}(x) \), then the kurtosis of \( x \) is defined as:

\[
\frac{N \sum (x_i - \text{mean}(x))^4 (\sum (x_i - \text{mean}(x))^2)^2 (\text{mean}(x))^2)}{(N-1)(N-2)(N-3)} - \frac{3(N-1)^2}{(N-2)(N-3)}
\]

Value

The kurtosis of \( x \).

References

Examples

\begin{verbatim}
x <- rnorm(100)
kurtosis(x)
\end{verbatim}

\textbf{npde}

Compute normalised prediction distribution errors

\section*{Description}

These functions compute normalised prediction distribution errors (npde) and optionally prediction discrepancies (pd). npde asks the user the name and structure of the files containing the data, using pdemenu, while autonpde takes these variables and others as arguments.

\section*{Usage}

\begin{verbatim}
autonpde(namobs, namsim, iid, ix, iy, imdv = 0, icens = 0, icov = 0, iipred = 0, boolsave = TRUE, namsav = "output", type.graph = "eps", verbose = FALSE, calc.npde=TRUE, calc.pd=TRUE, decorr.method = "cholesky", cens.method = "cdf", units = list(x="",y=""), detect=FALSE, ties=TRUE)
\end{verbatim}

\section*{Arguments}

\begin{itemize}
  \item \textbf{namobs} name of the file containing the observed data, or a dataframe containing the observed data (in both cases, the column containing the various data required for the computation of the pde can be set using the arguments iid, ix and iy below)
  \item \textbf{namsim} name of the file containing the simulated data, or a dataframe containing the simulated data (the program will assume that subject ID are in column 1 and simulated Y in column 3, see User Guide)
  \item \textbf{iid} name/number of the column in the observed data containing the patient ID; if missing, the program will attempt to detect a column named id
  \item \textbf{ix} name/number of the column in the observed data containing the independent variable (X); if missing, the program will attempt to detect a column named X
  \item \textbf{iy} name/number of the column in the observed data containing the dependent variable (Y); if missing, the program will attempt to detect a column with the response
  \item \textbf{imdv} name/number of the column containing information about missing data (MDV), defaults to 0 (column not present)
  \item \textbf{icens} name/number of the column containing information about censored data (cens), defaults to 0 (column not present)
  \item \textbf{icov} name/number of the column(s) containing covariate information defaults to 0 (no covariates)
\end{itemize}
**iipred**

name/number of the column(s) with individual predictions (ipred), defaults to 0 (individual predictions not available)

**units**
a list with components x, y and cov (optional), specifying the units respectively for the predictor (x), the response (y), and the covariates (a vector of length equal to the number of covariates). Units will default to (-) if not given.

**detect**
a boolean controlling whether automatic recognition of columns in the dataset is on, defaults to FALSE

**boolsave**
a boolean (TRUE if graphs and results are to be saved to a file, FALSE otherwise), defaults to TRUE

**namsav**
name of the files to which results are to be saved (defaults to "output", which will produce a file called output.eps (if the default format of postscript is kept, see type.graph) for the graphs and a file called output.npde for the numerical results (see value)

**type.graph**
type of graph (one of "eps","jpeg","png","pdf"), defaults to postscript ("eps")

**calc.npde**
a boolean (TRUE if npde are to be computed, FALSE otherwise), defaults to TRUE

**calc.pd**
a boolean (TRUE if pd are to be computed, FALSE otherwise), defaults to TRUE

**cens.method**
a character string indicating the method used to handle censored data (see npde.cens.method) defaults to cdf

**decorr.method**
a character string indicating the method used to decorrelate observed and simulated data in the computation of npde (see npde.decorr.method) defaults to cholesky

**ties**
a boolean (if FALSE, the distributions of pd and npde are smoothed by jittering the values so that there are no ties), defaults to TRUE

**verbose**
a boolean (TRUE if messages are to be printed as each subject is processed, FALSE otherwise), defaults to FALSE

**Details**

Both functions compute the normalised prediction distribution errors (and/or prediction discrepancies) in the same way. npde is an interactive function whereas autonpde takes all required input as arguments.

When the computation of npde fails because of numerical problems, error messages are printed out, then pd are computed instead and graphs of pd are plotted so that the user may evaluate why the computation failed.

The function also prints out the characteristics of the distribution of the npde (mean, variance, skewness and kurtosis) as well as the results of the statistical tests applied to npde. In addition, if boolsave is TRUE, two files are created:

- **results file** the numerical results are saved in a file with extension .npde (the name of which is given by the user). The file contains the components id, xobs, ypred, npde, pd stored in columns
- **graph file** the graphs are saved to a file with the same name as the results file, and with extension depending on the format.
Value

An object of class `NpdeObject`

Author(s)

Emmanuelle Comets <emmanuelle.comets@bichat.inserm.fr>

References


See Also

`npde.graphs`, `gof.test`

Examples

data(theopp)
data(simtheopp)

# Calling autonpde with dataframes
x<-autonpde(theopp,simtheopp,1,3,4,boolsave=FALSE)
x

# Calling autonpde with names of files to be read from disk
write.table(theopp,"theopp.tab",quote=FALSE,row.names=FALSE)
write.table(simtheopp,"simtheopp.tab",quote=FALSE,row.names=FALSE)
x<-autonpde(namobs="theopp.tab",namsim="simtheopp.tab",iid = 1,
ix = 3, iy = 4, imdv=0, boolsave = FALSE)
head(x["results"["res"]])

---

**npde.cens.method**  
*Method used to handle censored data*

**Description**

Specifies the method used to handle censored data (data below the limit of quantification LOQ)

**Details**

More details can be found in the PDF documentation.
Value

The following methods are available in the npde library:

**omit**  pd and npde for censored data will be set to NA

**cdf**  for an observation ycens_ij under the LOQ, a pd_ij will be imputed in the uniform distribution [0-pLOQ_ij] where pLOQ_ij is the probability that y_ij is below LOQ, according to the model; the predictive distribution will then be used to obtain a corresponding y*_ij. This is also performed for all simulated data, and the npde are then computed on the completed dataset containing the observed y_ij for the uncensored data and the y*_ij imputed for the censored data. This method is the default.

**ipred**  an observation ycens_ij is replaced by the individual prediction according to the model (ipred, which must be present in the dataset). Simulated data are left untouched.

**ppred**  an observation ycens_ij is replaced by the population prediction according to the model. Simulated data are left untouched.

**loq**  an observation ycens_ij is replaced by the value of the LOQ. Simulated data are left untouched.

References


---

npde.decorr.method  Method used to decorrelate vectors

Description

Specifies the method used to decorrelate observed and simulated data

Details

More details can be found in the PDF documentation.

Value

Decorrelation requires computing the square root of the inverse of the individual variance-covariance matrix Vi. The following methods are available in the npde library:

**cholesky**  decorrelation is performed through the Cholesky decomposition (default)

**inverse**  decorrelation is performed by inverting Vi through the eigen function

**polar**  the singular-value decomposition (svd) is used

References

Save the graphs for a NpdeObject object to a file

**Description**

Save the graphs to a file on disk

**Arguments**

- `object` a NpdeObject object
- `...` optional arguments to replace options in object

**Details**

The following options can be changed by passing the appropriate arguments: `namsav` (string giving the root name of the files, an extension depending on the type of graph will be added), `namgr` (string giving the full name of the file), `type.graph` (one of "eps", "pdf", "jpeg", "png")

**References**


Save the results contained in a NpdeObject object to a file

**Description**

Save the results to a table on disk

**Arguments**

- `object` a NpdeObject object
- `...` optional arguments to replace options in object

**Details**

The following options can be changed by passing the appropriate arguments: `namsav` (string giving the root name of the files, an extension `.npde` will be added), `nameres` (string giving the full name of the file)

**References**

npdeControl

Set options for an NpdeObject

Description

Set, replace and check options for an NpdeObject

Usage

npdeControl(boolsave = TRUE, namsav = "output",
        type.graph = "eps", verbose = FALSE, calc npde = TRUE,
        calc pd = TRUE, decorr.method = "cholesky", cens.method
        = "omit", ties = TRUE, sample = FALSE)

Arguments

boolsave whether to save the results (a file containing the numerical results and a file with

   the graphs)

namsav the root name of the files to save to (the file with the results will be named

   ROOTNAME.npde and the graphs will be saved to ROOTNAME.format where

   format is given by the type.graph argument)

type.graph type of graph to save to (one of "eps", "pdf", "jpeg", "png")

verbose a boolean; if TRUE, a message is printed as the computation of the npde begins

   for each new subject

calc pd a boolean; TRUE to compute pd

calc npde a boolean; TRUE to compute npde
decorr.method the method used to decorrelate simulated and observed data (see npde.decorr.method)
cens.method the method used to handle censored data (see npde.cens.method)
ties if FALSE, a smoothing will be applied to prediction discrepancies to avoid ties

sample if TRUE, the test on the pd will be performed after randomly sampling only pd

   per subject

npdeData

Creates a NpdeData object

Description

This function is used to create a NpdeData object, representing a longitudinal data structure, and fill

it with data from a dataframe or a file on disk
Usage

npdeData(name.data, header=TRUE, sep=",", na.strings=c(".", "NA"), name.group, name.predictor, name.response, name.covariates, name.cens, name.miss, name.ipred, units=list(x="", y="", covariates=c()), detect=TRUE, verbose=FALSE)

Arguments

name.data name of the file containing the observed data, or a dataframe containing the observed data
sep field separator (for files on disk)
na.strings strings to be considered as indicating NA
header boolean indicating whether the file has a header (mandatory if detect is TRUE)
name.group name/number of the column in the observed data containing the patient ID (if missing and detect is TRUE, columns named id, subject or sujet (regardless of case) will be assumed to contain this information)
name.predictor name/number of the column in the observed data containing the independent variable X (if missing and detect is TRUE, columns named xobs, time, dose, x, temps, tim (regardless of case) will be assumed to contain this information)
name.response name/number of the column in the observed data containing the dependent variable Y (if missing and detect is TRUE, columns named yobs, response, resp, conc, concentration (regardless of case) will be assumed to contain this information)
name.miss name/number of the column containing information about missing data (MDV) (if missing and detect is TRUE, column called mdv or miss (regardless of case) will be assumed to contain this information)
name.cens name/number of the column containing information about censored data (cens) (if missing and detect is TRUE, column with a name containing cens (regardless of case) will be assumed to contain this information)
name.covariates name/number of the column(s) containing covariate information (optional)
name.ipred name/number of the column(s) containing individual predictions (ipred) (if missing and detect is TRUE, column containing ipred (regardless of case) will be assumed to contain this information)
units a list with components x, y and cov (optional), specifying the units respectively for the predictor (x), the response (y), and the covariates (a vector of length equal to the number of covariates). Units will default to (-) if not given.
detect a boolean controlling whether automatic recognition of columns in the dataset is on, defaults to TRUE
verbose whether to print warning messages, defaults to FALSE (set to TRUE to check how data is being handled)

Value

an object of class NpdeData
Author(s)
Emmanuelle Comets <emmanuelle.comets@bichat.inserm.fr>

References

See Also
npde, autonpde

Examples
data(theopp)

x<-npdedata(theopp) # Automatic detection
print(x)

x<-npdedata(theopp,name.group="ID", name.predictor="Time", name.response="Conc",
name.covariates=c("Wt"), units=list(x="hr", y="mg/L", covariates="kg")) # Explicit
print(x)
plot(x)

NPdeData-class  
Class "NPdeData" representing the structure of the longitudinal data

Description
A longitudinal data structure

Objects from the Class
NPdeData objects are typically created by calls to npdeData and contain the following slots:

- **name.data** character string giving the name of the dataset
- **name.group** character string giving the name of the grouping term (ID)
- **name.predictor** character string giving the name of the predictor (X)
- **name.response** character string giving the name of the response (Y)
- **name.cens** character string giving the name of the censoring indicator
- **name.mdv** character string giving the name of the missing data indicator
- **name.covariates** vector of character string giving the name(s) of the covariates
- **name.ipred** character string giving the name of the individual predictions
- **units** (optional) a list with the units for X, Y, and covariates
- **data** a dataframe containing the data
NpdeObject-class

N number of subjects
ntot.obs total number of non-missing observations
nind.obs vector of size N giving the number of non-missing observations for each subject
ind index of non-missing observations
icens index of censored observations (non-missing)
not.miss a vector of boolean indicating for each observation whether it is missing (FALSE) or available (TRUE)
loq the censoring value

Methods

npdeData(name.data): Create a new NpdeData object from dataset name.data
print(npde.data): Prints a summary of object npde.data
show(npde.data): Prints a short summary of object npde.data
showall(npde.data): Prints a detailed summary of object npde.data
plot(npde.data): Plots the data in npde.data. More details can be found in plot.NpdeData
summary(npde.data): Returns a summary of object npde.data in list format
set.plotoptions(npde.data): Sets options for graphs of npde.data (internal method used in plots)

See Also

npde, autonpde, plot.NpdeData

Examples

methods(class="NpdeData")
showClass("NpdeData")

NpdeObject-class Class "NpdeObject"

Description

An object of class NpdeObject

Objects from the Class

NpdeObject objects are typically created by calls to npde or autonpde. They contain the following slots:

data an object of class NpdeData, containing the observed data
sim.data an object of class NpdeSimData, containing the simulated data
res an object of class NpdeRes, containing the results
options a list of options
prefs a list of graphical preferences for the plots
Methods

\textbf{print(\textit{x})}: Prints a summary of object  
\textbf{show(\textit{x})}: Prints a short summary of object  
\textbf{showall(\textit{x})}: Prints a detailed summary of object  
\textbf{plot(\textit{x})}: Diagnostic and other plots. More details can be found in \texttt{plot.NpdeObject}  
\textbf{summary(\textit{x})}: Returns a summary of object \textit{x} in list format  
\textbf{gof.test(\textit{x}, which="npde", parametric=TRUE, ...)}: Returns goodness-of-fit tests  
\textbf{set.plot.options(\textit{x})}: Sets options for graphs (internal method used in plots)

See Also

\texttt{npde, autonpde, NpdeData, NpdeSimData, NpdeRes, gof.test}

Examples

```r
methods(class="NpdeObject")
showClass("NpdeObject")
```

---

**NpdeSimData-class**  
Class "NpdeSimData" representing the structure of the longitudinal data

Description

A longitudinal data structure, with simulated data

Objects from the Class

NpdeSimData objects are created by associating an NpdeData object with matching simulated data, and they contain the following slots.

- \texttt{name.simdata} character string giving the name of the dataset  
- \texttt{nrep} number of replications  
- \texttt{datsim} a dataframe containing the simulated data, with columns: idsim (subject id), irsim (replication index), xsim (simulated x), ysim (simulated response). After a call to \texttt{npde} or \texttt{autonpde}, an additional column ydsim (decorrelated replicated data) will be added.

Methods

- \textbf{print(\texttt{npde.simdata})}: Prints a summary of object \texttt{npde.simdata}  
- \textbf{show(\texttt{npde.simdata})}: Prints a short summary of object \texttt{npde.simdata}  
- \textbf{showall(\texttt{npde.simdata})}: Prints a detailed summary of object \texttt{npde.simdata}
Plot the data in a `NpdeData` object.

**Usage**

```r
## S3 method for class 'NpdeData'
plot(x, y, ...)
```

**Arguments**

- `x`: a `NpdeData` object
- `y`: unused, here for compatibility with the base plot function
- `...`: additional graphical parameters to be passed on to the plot

**Details**

The default plot is a spaghetti plot of all the data, with a line joining the observations for each subject. If censored data is present, it is shown with a different symbol and colour.

**References**


**See Also**

- `set.plotoptions`

**Examples**

```r
data(theopp)

x <- npdeData(theopp, name.group="ID", name.predictor="Time", name.response="Conc", name.covariates=c("Wt"), units=list(x="hr", y="mg/L", covariates="kg"))

plot(x)
```
plot.NpdeObject

Plots a NpdeObject object

Description
Plots the data and diagnostic plots in a NpdeObject object

Usage
### S3 method for class 'NpdeObject'
plot(x, y, ...)

Arguments
- `x` a NpdeObject object
- `y` unused, here for compatibility with the base plot function
- `...` additional graphical parameters, which when given will supersede graphical preferences stored in the object

Details
The default plot

References

See Also
- `set.plotoptions`

Examples
```r
data(theopp)
data(simtheopp)
x<-autonpde(theopp,simtheopp,iid="ID",ix="Time", iy="Conc", boolsave=FALSE)
plot(x)
```
set.plotoptions

Set graphical preferences

Description
This function is used to set options for graphs

Usage
set.plotoptions(object, ...)

Arguments
object an object of class NpdeData or NpdeObject
... arguments to replace default arguments (currently ignored)

Details
See documentation for a list of available options.

Value
a list of options for graphs

Author(s)
Emmanuelle Comets <emmanuelle.comets@bichat.inserm.fr>

See Also
npde, autonpde

showall

Brief summary of an object

Description
Prints a brief summary of an object

Usage
showall.NpdeData(object)

Arguments
object an object
Simulated data for the computation of normalised prediction distribution errors

Description

The `simtheopp` dataset contains 100 simulations using the design of dataset `theopp`. These simulations are used to compute npde. The control file used to perform the simulations can be found in the subdirectory 'doc' within the library npde.

Usage

`simtheopp`

Format

This data frame contains the following columns:

- **ID** an ordered factor with levels 1, . . . , 12 identifying the subject on whom the observation was made. The ordering is first by simulation then by increasing time.
- **xsim** time since drug administration when the sample was drawn (hr).
- **ysim** simulated theophylline concentration (mg/L).

Details

See `theopp` for a description of the original dataset.

The simulated data was obtained using the software NONMEM. A one-compartment model was fit to the data. An exponential interindividual variability was assumed for the three parameters (absorption rate constant ka, volume of distribution V and clearance CL) and a combined additive and proportional residual error model was used. The estimated parameters were then used to simulate 100 datasets with the same structure as the original dataset. Thus, for each observation in the original dataset, the simulated dataset contains 100 simulations under the model used for the estimation.

This dataset is provided so that users can figure out what type of data is needed for the computation of prediction distribution errors. More information can be found in the User Guide distributed along with this package, which contains a run-through of the theophylline example.

Source


See Also

`theopp`
Examples

data(simtheopp)

# Plotting the simulated data for subject 1 in the first simulation
plot(ysim[2:12]-xsim[2:12], data=simtheopp, xlab="Time after dose (hr)", ylab="Theophylline concentration (mg/L)", type="l", main="Example of simulated data for subject 1")

# Plotting a 90% prediction interval for the observations in theopp
# using the simulated data in simtheopp
# note: differences in doses between subjects are not taken into account
data(theopp)
xpl<-c(0,0.25,0.5,1,2,3,5,7,9,12,24)
xpl1<-list(c(0,0.1),c(0.2,0.4),c(0.5,0.65),c(0.9,1.2),c(1.9,2.2),c(3.4,4),
c(4.9,5.2),c(6.9,7.2),c(8.8,9.4),c(11.5,12.2),c(23.7,24.7))
ypl<-cbind(xpl=xpl,binf=xpl,median=xpl,bsup=xpl)
for(i in 1:(length(xpl))) {
  vec<-simtheopp$ysim[simtheopp$xsim=xpl1[[i]][1] & simtheopp$xsim=xpl1[[i]][2]]
  ypl[i,2:4]<-quantile(vec,c(0.05,0.5,0.95))
}
plot(Conc~Time, data=theopp, xlab="Time after dose (hr)", ylab="Theophylline concentration (mg/L)")
lines(ypl[,1],ypl[,3],lwd=2)
lines(ypl[,1],ypl[,2],lty=2)
lines(ypl[,1],ypl[,4],lty=2)

---

simvirload Simulated data for the computation of normalised prediction distribution errors, viral load example

Description

The simvirload dataset contains 1000 simulations using the design of dataset virload. These simulations are used to compute npde.

Usage

simvirload

Format

This data frame contains the following columns:

- **ID** an ordered factor with levels 1, ..., 50 identifying the subject on whom the observation was made. The ordering is first by simulation then by increasing time.
- **xsim** time (day).
- **ysim** simulated viral loads, in base 10 log-scale (cp/L).
Details

See virload for a description of the original dataset.

The simulated data was obtained using the software R, as described in Nguyen et al. (2011).

Source


Nguyen, T., Comets, E., Mentre, F. (2010). Prediction discrepancies (pd) for evaluation of models with data under limit of quantification. 20th meeting of the population approach group in Europe (PAGE), Athens, Greece. Abstr 2182.

See Also

virload

<table>
<thead>
<tr>
<th>skewness</th>
<th>Skewness</th>
</tr>
</thead>
</table>

Description

Computes the skewness.

Usage

skewness(x)

Arguments

x a numeric vector containing the values whose skewness is to be computed. NA values are removed in the computation.

Details

If \( N = \text{length}(x) \), then the skewness of \( x \) is defined as

\[
N^{-1/3} \text{sd}(x)^{-3} \sum (x_i - \text{mean}(x))^3.
\]

Value

The skewness of \( x \).
theopp

References


Examples

```r
x <- rnorm(100)
skewness(x)
```

---

### theopp

**Pharmacokinetics of theophylline**

**Description**

The theopp data frame has 132 rows and 5 columns of data from an experiment on the pharmacokinetics of theophylline.

**Usage**

theopp

**Format**

This data frame contains the following columns:

- **ID** an ordered factor with levels 1, ..., 12 identifying the subject on whom the observation was made. The ordering is by Time at which the observation was made.
- **Dose** dose of theophylline administered orally to the subject (mg).
- **Time** time since drug administration when the sample was drawn (hr).
- **Conc** theophylline concentration in the sample (mg/L).
- **Wt** weight of the subject (kg).

**Details**

Boeckmann, Sheiner and Beal (1994) report data from a study by Dr. Robert Upton of the kinetics of the anti-asthmatic drug theophylline. Twelve subjects were given oral doses of theophylline then serum concentrations were measured at 11 time points over the next 25 hours. In the present package *npde*, we removed the data at time 0.

These data are analyzed in Davidian and Giltinan (1995) and Pinheiro and Bates (2000) using a two-compartment open pharmacokinetic model.

These data are also available in the library datasets under the name Theoph in a slightly modified format and including the data at time 0. Here, we use the file in the format provided in the *NONMEM* installation path (see the User Guide for that software for details).
Source


Examples
data(theopp)
str(theopp)

#Plotting the theophylline data
plot(Conc~Time,data=theopp,xlab="Time after dose (hr)",
ylab="Theophylline concentration (mg/L)")

virload Simulated HIV viral loads in HIV patients

Description
This is simulated data, based on real data obtained in a phase II clinical trial supported by the French Agency for AIDS Research, the COPHAR 3-ANRS 134 trial (Goujard et al., 2010). The original study included 35 patients, who received a once daily dose containing atazanavir (300 mg), ritonavir (100 mg), tenofovir disoproxil (245 mg) and emtricitabine (200 mg) during 24 weeks. Viral loads were measured 6 times over a treatment period of 24 weeks (day 0, 28, 56, 84, 112, 168).

The datasets were generated in a simulation study designed to evaluate the new method proposed to handle BQL data (Nguyen et al., 2011). Data was simulated using a simple bi-exponential HIV dynamic model describing the two-phase decline of viral load during anti-retroviral treatment.

The virload data frame has 300 rows and 4 columns of data. The dataset was then censored at two different LOQ levels (LOQ=20 or 50~copies/mL) to generate two datasets containing different proportions of BQL data, creating the data frames virload20 and virload50 respectively.

Usage

virload

Format
This data frame contains the following columns:

ID an ordered factor with levels 1, . . . , 50 identifying the subject on whom the observation was made. The ordering is by Time at which the observation was made.

Time time since the beginning of the study (days).
**Log_VL**  logarithm (base 10) of the viral load (copies/L).

**cens**  indicator variable (cens=1 for censored data, cens=0 for observed data)

**ipred**  individual predictions)

**Source**


Nguyen, T., Comets, E., Mentre, F. (2010). Prediction discrepancies (pd) for evaluation of models with data under limit of quantification. 20th meeting of the population approach group in Europe (PAGE), Athens, Greece. Abstr 2182.

**Examples**

data(virload)
str(virload)
data(virload50)

#Plotting the data
plot(Log_VL~Time,data=virload,xlab="Time (d)",ylab="Viral loads, base 10 logarithmic scale (cp/mL)")
plot(Log_VL~Time,data=virload50,xlab="Time (d)",ylab="Viral loads, base 10 logarithmic scale (cp/mL)")
Index

*Topic IO
  npde_graphs, 11
  npde_save, 11
*Topic classes
  NpdeData-class, 14
  NpdeObject-class, 15
  NpdeSimData-class, 16
*Topic datasets
  simtheopp, 20
  simvirload, 21
  theopp, 23
  virload, 24
*Topic files
  npde_graphs, 11
  npde_save, 11
*Topic methods
  npde.cens.method, 9
  npde.decorr.method, 10
  npdeControl, 12
*Topic models
  npde, 7
  npde-package, 2
  npdeData, 12
*Topic plot
  plot.NpdeData, 17
  plot.NpdeObject, 18
  set.plotoptions, 19
*Topic print
  showall, 19
*Topic test
  gof.test, 4
*Topic univar
  kurtosis, 6
  skewness, 22
  [,NpdeData-method (NpdeData-class), 14
  [,NpdeObject-method (NpdeObject-class), 15
  [,NpdeSimData-method (NpdeSimData-class), 16
  <-,NpdeData-method (NpdeData-class), 14
  <-,NpdeObject-method (NpdeObject-class), 15
  <-,NpdeSimData-method (NpdeSimData-class), 16
  autonpde, 3, 4, 14–17, 19
  autonpde (npde), 7
  calcnpde.sim(dist.pred.sim), 3
  check.control.options (npdeControl), 12
  dist.pred.sim, 3
  gof.test, 4, 9, 16
  kurtosis, 5, 6, 6
  na.action, 5
  npde, 3, 4, 7, 14–17, 19
  npde-package, 2
  npde.cens.method, 8, 9, 12
  npde.decorr.method, 8, 10, 12
  npde.graphs, 9, 11
  npde.graphs,NpdeObject (NpdeObject-class), 15
  npde.graphs,NpdeObject-method (npde.graphs), 11
  npde.main,NpdeObject (NpdeObject-class), 15
  npde.save, 11
  npde.save,NpdeObject (NpdeObject-class), 15
  npde.save,NpdeObject-method (npde.save), 11
  npdeControl, 12
  NpdeData, 15, 16
  NpdeData (NpdeData-class), 14
  npdeData, 12, 14
  NpdeData-class, 14
  NpdeObject, 9
INDEX

npdeobject (npdeobject-class), 15
npdeobject-class, 15
npdeobject-class, (npdeobject-class), 15
npdeRes, 16
npdeSimData, 16
npdeSimData (npdeSimData-class), 16
npdeSimData-class, 16
plot, npdeObject (npdeObject-class), 15
plot, npdeData, 15, 17
plot, npdeObject, 16, 18
print, npdeData-method (npdeData-class), 14
print, npdeObject-method (npdeObject-class), 15
print.gof.test (gof.test), 4
replace.control.options (npdeControl), 12
set.plotoptions, 17, 18, 19
set.plotoptions, npdeData-method (set.plotoptions), 19
set.plotoptions, npdeObject-method (set.plotoptions), 19
set.plotoptions, npdeData (set.plotoptions), 19
show, npdeData-method (npdeData-class), 14
show, npdeObject-method (npdeObject-class), 15
show, npdeSimData-method (npdeSimData-class), 16
showall, 19
showall, npdeData-method (showall), 19
showall, npdeObject-method (npdeObject-class), 15
showall, npdeSimData-method (showall), 19
showall, npdeData (showall), 19
showall, npdeObject (showall), 19
showall, npdeRes (showall), 19
showall, npdeSimData (showall), 19
simtheopp, 20
simvirload, 21
skewness, 5, 6, 22
summary, npdeData-method (npdeData-class), 14
summary, npdeObject-method (npdeObject-class), 15
test, npdeObject-method (npdeObject-class), 15
theopp, 20, 23
virload, 21, 22, 24
virload20 (virload), 24
virload50 (virload), 24