

Package ‘nlstools’

May 6, 2012

Version 0.0-12

Date Fri Apr 16:58:00 CET 2012

Title Tools for nonlinear regression diagnostics

Author Florent Baty <florent.baty@gmail.com> and Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>, with contributions from Sandrine Charles, Jean-Pierre Flandrois

Maintainer Florent Baty <florent.baty@gmail.com>

Depends stats

Description Several tools for assessing the quality of fit of a gaussian nonlinear model are provided.

License GPL (>= 2)

Repository CRAN

Repository/R-Forge/Project nlstools

Repository/R-Forge/Revision 5

Date/Publication 2012-05-06 07:06:10

R topics documented:

competitioncurve	2
competitionmodels	3
growthcurve	5
growthmodels	6
michaelisdata	8
michaelismodels	9
nlsBoot	10
nlsConfRegions	12
nlsContourRSS	14
nlsJack	15
nlsResiduals	17

nlstools	18
ross	20
secondary	21
survivalcurve	23
survivalmodels	24

Index	27
--------------	-----------

competitioncurve	<i>Growth kinetics of two competitive bacterial flora</i>
------------------	---

Description

Kinetics of simultaneous growth of *Escherichia coli* O157:H7 and ground beef background microflora in enrichment broth

Usage

```
data(competition1)
data(competition2)
```

Format

Data frames with 3 columns (t: time, flora : 1 for the first flora and 2 for the second one, LOG10N: decimal logarithm of bacterial density)

Source

Two of the kinetics used in Vimont et al. (2006)

References

Vimont A, Vernozzy-Rozand C, Montet MP, Lazizzera C, Bavai C and Delignette-Muller ML (2006) Modeling and predicting the simultaneous growth of *Escherichia coli* O157:H7 and ground beef background microflora in various enrichment protocols. *Applied and Environmental Microbiology* **72**, 261-268.

Examples

```
data(competition1)
data(competition2)
def.par <- par(no.readonly = TRUE)
par(mfrow = c(1,2))
twocolors <- c("red", "blue")
plot(competition1$t, competition1$LOG10N, col=twocolors[competition1$flora])
plot(competition2$t, competition2$LOG10N, col=twocolors[competition2$flora])
par(def.par)
```

Description

Formulas of primary growth models used in predictive microbiology to model the simultaneous growth of two competitive bacterial flora assuming a Jameson effect

Usage

jameson_buchanan
jameson_baranyi
jameson_without_lag

Details

These models describe the simultaneous evolution of the decimal logarithm of the microbial counts of two flora (LOG10N) as a function of the time (t) and of the flora (flora) coded as 1 for counts of flora 1 and 2 for counts of flora 2. These three models assume independent lag and growth parameters for flora 1 and 2, except for the saturation which is supposed to be governed by the Jameson effect and modelled by a common parameter (tmax) which represents the time at which both flora stop to multiply. Modelling the simultaneous saturation by this way enables the model to be fitted by nls, as an analytical form of the model is available.

jameson_buchanan is based on the model of Buchanan et al. (1997) for lag phase modelling and is characterized by seven parameters (LOG10N0_1, mumax_1, lag_1, LOG10N0_2, mumax_2, lag_2 and the common saturation time tmax). This model was described and used in Vimont et al. (2006).

jameson_baranyi is based on the model of Baranyi and Roberts (1994) for lag phase modelling and is characterized by seven parameters (LOG10N0_1, mumax_1, lag_1, LOG10N0_2, mumax_2, lag_2 and the common saturation time tmax)

jameson_without_lag is based on the exponential model without lag phase and is thus characterized by five parameters (LOG10N0_1, mumax_1, LOG10N0_2, mumax_2 and the common saturation time tmax)

Value

A formula

Author(s)

Florent Baty <florent.baty@gmail.com>
Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Baranyi J and Roberts, TA (1994) A dynamic approach to predicting bacterial growth in food, *International Journal of Food Microbiology*, **23**, 277-294.

Buchanan RL, Whiting RC, Damert WC (1997) When is simple good enough: a comparison of the Gompertz, Baranyi, and three-phase linear models for fitting bacterial growth curves. *Food Microbiology*, **14**, 313-326.

Vimont A, Vernozy-Rozand C, Montet MP, Lazizzera C, Bavai C and Delignette-Muller ML (2006) Modeling and predicting the simultaneous growth of *Escherichia coli* O157:H7 and ground beef background microflora in various enrichment protocols. *Applied and Environmental Microbiology* **72**, 261-268.

Examples

```
options(digits=3)

### Example 1: fit of model jameson_buchanan
data(competition1)
nls1 <- nls(jameson_buchanan, competition1,
            list(lag_1 = 2, mumax_1 = 1, LOG10N0_1 = 1, tmax = 12,
                 lag_2 = 2, mumax_2 = 1, LOG10N0_2 = 4))

overview(nls1)

# Plot of theoretical curves with data
twocolors <- c("red", "blue")
npoints <- 100
seq.t <- seq(0, max(competition1$t), length.out=npoints)
prednls1.1 <- predict(nls1, data.frame(t=seq.t, flora=rep(1, npoints)))
prednls1.2 <- predict(nls1, data.frame(t=seq.t, flora=rep(2, npoints)))
plot(competition1$t, competition1$LOG10N, col=twocolors[competition1$flora], xlab="t", ylab="LOG10N")
lines(seq.t, prednls1.1, col=twocolors[1])
lines(seq.t, prednls1.2, col=twocolors[2])

### Example 2 : fit of model jameson_baranyi
data(competition1)
nls2 <- nls(jameson_baranyi, competition1,
            list(lag_1 = 2, mumax_1 = 1, LOG10N0_1 = 1, tmax = 12,
                 lag_2 = 2, mumax_2 = 1, LOG10N0_2 = 4))

overview(nls2)
plotfit(nls2)

# Plot of theoretical curves with data
twocolors <- c("red", "blue")
npoints <- 100
seq.t <- seq(0, max(competition1$t), length.out=npoints)
prednls2.1 <- predict(nls2, data.frame(t=seq.t, flora=rep(1, npoints)))
prednls2.2 <- predict(nls2, data.frame(t=seq.t, flora=rep(2, npoints)))
```

```

plot(competition1$t,competition1$LOG10N,col=twocolors[competition1$flora],xlab="t",ylab="LOG10N")
lines(seq.t,prednls2.1,col=twocolors[1])
lines(seq.t,prednls2.2,col=twocolors[2])

### Example 3: fit of model jameson_without_lag
data(competition2)
nls3 <- nls(jameson_without_lag, competition2,
  list(mumax_1 = 1, LOG10N0_1 = 1, tmax = 12,
  mumax_2 = 1, LOG10N0_2 = 4))

overview(nls3)
plotfit(nls3)

# Plot of theoretical curves with data
twocolors <- c("red","blue")
npoints <- 100
seq.t <- seq(0,max(competition2$t),length.out=npoints)
prednls3.1 <- predict(nls3,data.frame(t=seq.t,flora=rep(1,npoints)))
prednls3.2 <- predict(nls3,data.frame(t=seq.t,flora=rep(2,npoints)))
plot(competition2$t,competition2$LOG10N,col=twocolors[competition2$flora],xlab="t",ylab="LOG10N")
lines(seq.t,prednls3.1,col=twocolors[1])
lines(seq.t,prednls3.2,col=twocolors[2])

```

growthcurve

Bacterial kinetics data sets

Description

Bacterial kinetics data sets

Usage

```

data(growthcurve1)
data(growthcurve2)
data(growthcurve3)
data(growthcurve4)

```

Format

Data frames with 2 columns (t: time, LOG10N: decimal logarithm of bacterial density)

Source

Data obtained by Florent Baty <florent.baty@gmail.com> and Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

Examples

```
data(growthcurve1)
data(growthcurve2)
data(growthcurve3)
data(growthcurve4)
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2,2))
plot(growthcurve1)
plot(growthcurve2)
plot(growthcurve3)
plot(growthcurve4)
par(def.par)
```

growthmodels

Bacterial growth models

Description

Formulas of primary growth models commonly used in predictive microbiology

Usage

```
baranyi
baranyi_without_Nmax
baranyi_without_lag
buchanan
buchanan_without_Nmax
buchanan_without_lag
gompertz
```

Details

These models describe the evolution of the decimal logarithm of the microbial count (LOG₁₀N) as a function of the time (t).

baranyi is the model of Baranyi and Roberts (1994) with four parameters (LOG₁₀N₀, μ_{max}, lag, LOG₁₀N_{max})

baranyi_without_Nmax is the model of Baranyi and Roberts (1994) with three parameters (LOG₁₀N₀, μ_{max}, lag), without braking

baranyi_without_lag is the model of Baranyi and Roberts (1994) with three parameters (LOG₁₀N₀, μ_{max}, LOG₁₀N_{max}), without lag

buchanan is the three-phase linear model proposed by Buchanan et al. (1997)

buchanan_without_Nmax is the two-phase linear model with three parameters (LOG₁₀N₀, μ_{max}, lag), without braking

buchanan_without_lag is the two-phase linear model with three parameters (LOG10N0, mumax, LOG10Nmax), without lag

gompertz is the modified Gompertz model introduced by Gibson et al. (1988) and reparameterized by Zwietering et al. (1990)

Value

A formula

Author(s)

Florent Baty <florent.baty@gmail.com>

Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Baranyi J and Roberts, TA (1994) A dynamic approach to predicting bacterial growth in food, *International Journal of Food Microbiology*, **23**, 277-294.

Buchanan RL, Whiting RC, Damert WC (1997) When is simple good enough: a comparison of the Gompertz, Baranyi, and three-phase linear models for fitting bacterial growth curves. *Food Microbiology*, **14**, 313-326.

Gibson AM, Bratchell N, Roberts TA (1988) Predicting microbial growth: growth responses of salmonellae in a laboratory medium as affected by pH, sodium chloride and storage temperature. *International Journal of Food Microbiology*, **6**, 155-178.

Zwietering MH, Jongenburger I, Rombouts FM, Van't Riet K (1990) Modeling of the bacterial growth curve. *Applied and Environmental Microbiology*, **56**, 1875-1881.

Examples

```
# Example 1

data(growthcurve1)
nls1 <- nls(baranyi, growthcurve1,
list(lag=4, mumax=1, LOG10N0 = 4, LOG10Nmax = 9))
nls2 <- nls(gompertz, growthcurve1,
list(lag = 4, mumax = 1, LOG10N0 = 4, LOG10Nmax = 9))
nls3 <- nls(buchanan, growthcurve1,
list(lag = 4, mumax = 1, LOG10N0 = 4, LOG10Nmax = 9))
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2,2))
plotfit(nls1, smooth = TRUE)
plotfit(nls2, smooth = TRUE)
plotfit(nls3, smooth = TRUE)
```

```
par(def.par)

# Example 2

data(growthcurve2)
nls4 <- nls(baranyi_without_Nmax, growthcurve2,
list(lag = 2, mumax = 0.4, LOG10N0 = 7.4))
nls5 <- nls(buchanan_without_Nmax, growthcurve2,
list(lag = 2, mumax = 0.4, LOG10N0 = 7.4))
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2,1))
plotfit(nls4, smooth = TRUE)
plotfit(nls5, smooth = TRUE)
par(def.par)

# Example 3

data(growthcurve3)
nls6 <- nls(baranyi_without_lag, growthcurve3,
list(mumax = 1, LOG10N0 = 0, LOG10Nmax = 5))
nls7 <- nls(buchanan_without_lag, growthcurve3,
list(mumax = 1, LOG10N0 = 0, LOG10Nmax = 5))
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2,1))
plotfit(nls6, smooth = TRUE)
plotfit(nls7, smooth = TRUE)
par(def.par)
```

michaelisdata

Michaelis Menten data sets

Description

Michaelis Menten data sets

Usage

```
data(vmkm)
data(vmkmki)
```

Format

vmkm is a data frame with 2 columns (S: concentration of substrat, v: volume)
vmkmki is a data frame with 3 columns (S: concentration of substrat, I: concentration of inhibitor,
v: volume)

Source

These datasets were provided by the French research unit INRA UMR1233.

Examples

```
data(vkm)
data(vkmki)
plot(vkm)
plot(vkmki)
```

michaelismodels	<i>Michaelis-Menten model and derived equations to model competitive and non-competitive inhibition</i>
-----------------	---

Description

Formula of Michaelis-Menten model commonly used to describe enzyme kinetics, and derived formulas taking into account the effect of a competitive or a non-competitive inhibitor

Usage

```
michaelis
compet_mich
non_compet_mich
```

Details

These models describe the evolution of the reaction rate (v) as a function of the concentration of substrate (S) and the concentration of inhibitor (I) for `compet_mich` and `non_compet_mich`.

`michaelis` is the classical Michaelis-Menten model (Dixon, 1979) with two parameters (K_m , V_{max}):

$$v = \frac{S}{S + K_m} V_{max}$$

`compet_mich` is the Michaelis-Menten derived model with three parameters (K_m , V_{max} , K_i), describing a competitive inhibition :

$$v = \frac{S}{S + K_m(1 + \frac{I}{K_i})} V_{max}$$

`non_compet_mich` is the Michaelis-Menten derived model with three parameters (K_m , V_{max} , K_i), describing a non-competitive inhibition :

$$v = \frac{S}{(S + K_m)(1 + \frac{I}{K_i})} V_{max}$$

Value

A formula

Author(s)

Florent Baty <florent.baty@gmail.com>

Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Dixon M and Webb EC (1979) *Enzymes*, Academic Press, New York.

Examples

```
# Example 1

data(vmkm)
nls1 <- nls(michaelis, vmkm, list(Km=1, Vmax=1))
plotfit(nls1, smooth = TRUE)

# Example 2

data(vmkmki)
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2,2))

nls2_c <- nls(compet_mich, vmkmki, list(Km=1, Vmax=20, Ki=0.5))
plotfit(nls2_c, variable=1)
overview(nls2_c)
res2_c <- nlsResiduals(nls2_c)
plot(res2_c, which=1)

nls2_nc <- nls(non_compet_mich, vmkmki, list(Km=1, Vmax=20, Ki=0.5))
plotfit(nls2_nc, variable=1)
overview(nls2_nc)
res2_nc <- nlsResiduals(nls2_nc)
plot(res2_nc, which=1)

par(def.par)
```

nlsBoot

Bootstrap resampling

Description

Bootstrap resampling

Usage

```
nlsBoot (nls, niter = 999)
## S3 method for class 'nlsBoot'
  plot(x, type = c("pairs", "boxplot"),
       mfr = c(ceiling(sqrt(ncol(x$coefboot))),
              ceiling(sqrt(ncol(x$coefboot)))),
       ask = FALSE, ...)
## S3 method for class 'nlsBoot'
  print(x, ...)
## S3 method for class 'nlsBoot'
  summary(object, ...)
```

Arguments

nls	an object of class 'nls'
niter	number of iterations
x, object	an object of class 'nlsBoot'
type	type of representation (options are "pairs" or "boxplot")
mfr	layout definition (number of rows and columns in the graphics device)
ask	if TRUE, draw plot interactively
...	further arguments passed to or from other methods

Details

Non-parametric bootstrapping is used. Mean centered residuals are bootstrapped. By default, 999 resampled data sets are created from which bootstrap estimates are obtained by fitting the model on each of these data sets. Whenever the fit fails to converge, a flag reports the number of non-convergences. If the fitting procedure fails to converge in more than 50% of the cases, the procedure is interrupted with a flag and no result is given. The function summary returns the bootstrap estimates and the 95 percent confidence intervals which are simply defined by the median and the 2.5 and 97.5 percentiles of the bootstrap sample of estimates. The bootstrap estimates distributions can be visualized using the function plot.nlsBoot either by plotting the bootstrap sample for each pair of parameters or by displaying the boxplot representation of the bootstrap sample for each parameter. Notice that nlsBoot does not currently handle transformed dependent variables specified in the left side of the nls formula.

Value

nlsBoot returns a list of three objects:

coefboot	contains the bootstrap parameter estimates
bootCI	contains the bootstrap medians and the bootstrap 95% confidence intervals
rse	is the vector of bootstrap residual errors

Author(s)

Florent Baty <florent.baty@gmail.com>
 Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Bates DM and Watts DG (1988) Nonlinear regression analysis and its applications. Wiley, Chichester, UK.

Huet S, Bouvier A, Poursat M-A, Jolivet E (2003) Statistical tools for nonlinear regression: a practical guide with S-PLUS and R examples. Springer, Berlin, Heidelberg, New York.

Examples

```
data(growthcurve4)
nls1 <- nls(gompertz, growthcurve4,
list(lag = 10, mumax = 0.1, LOG10N0 = 6, LOG10Nmax = 9))
boo <- nlsBoot(nls1)
plot(boo)
plot(boo, type = "boxplot", ask=FALSE)
summary.nlsBoot(boo)
```

nlsConfRegions

Confidence regions

Description

Draws parameter values in the Beale's 95 percent unlinearized confidence region

Usage

```
nlsConfRegions (nls, length = 1000, exp = 1.5)
## S3 method for class 'nlsConfRegions'
plot(x, bounds = FALSE, ask = FALSE, ...)
## S3 method for class 'nlsConfRegions'
print(x, ...)
```

Arguments

nls	an object of class 'nls'
length	number of points to draw in the confidence region
exp	expansion factor of the hypercube in which random values of parameters are drawn
x	an object of class 'nlsConfRegions'
bounds	logical defining whether bounds of the drawing hypercube are plotted
ask	if TRUE, draw plot interactively
...	further arguments passed to or from other methods

Details

A sample of points in the 95 percent confidence region is computed according to Beale's criterion (Beale, 1960). This region is also named the joint parameter likelihood region (Bates and Watts, 1988). The method used consists in a random sampling of parameters values in a hypercube centered on the least squares estimate and rejecting the parameters values whose residual sum of squares do not verify the Beale criterion. The confidence region is plotted by projection of the sampled points in each plane defined by a couple of parameters. Bounds of the hypercube in which random values of parameters are drawn may be plotted in order to check if the confidence region was totally included in the hypercube defined by default. If not the hypercube should be expanded in order to obtain the full confidence region

Value

nlsConfRegions returns a list of four objects:

cr	a data frame containing the sample drawn in the Beale's confidence region
rss	a vector containing the residual sums of squares corresponding to cr
rss95	the 95 percent residual sum of squares threshold according to Beale (1960)
bounds	lower and upper bounds of the hypercube in which random values of parameters have been drawn

Author(s)

Florent Baty <florent.baty@gmail.com>

Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Beale EML (1960) Confidence regions in non-linear estimations. *Journal of the Royal Statistical Society*, **22B**, 41-88.

Bates DM and Watts DG (1988) Nonlinear regression analysis and its applications. Wiley, Chichester, UK.

See Also

ellipse.nls in the ellipse library

Examples

```
data(growthcurve4)
nls1 <- nls(gompertz, growthcurve4,
list(lag = 10, mumax = 0.1, LOG10N0 = 6, LOG10Nmax = 9))
cr <- nlsConfRegions(nls1, exp = 2)
plot(cr, bounds = TRUE)
```

nlsContourRSS	<i>Surface contour of RSS</i>
---------------	-------------------------------

Description

Provides residual sum of squares (RSS) contours

Usage

```
nlsContourRSS (nls, lseq = 100, exp = 2)
## S3 method for class 'nlsContourRSS'
plot(x, nlev = 0, col = TRUE, col.pal = terrain.colors(100), ask = FALSE, useRaster = TRUE, ...)
## S3 method for class 'nlsContourRSS'
print(x, ...)
```

Arguments

nls	an object of class 'nls'
lseq	length of the sequences of parameters
exp	expansion factor of the parameter intervals defining the grids
nlev	number of contour levels to add to the likelihood contour at level 95 percent
col	logical. Contours are plotted with colors if TRUE
col.pal	Palette of colors. Colors to be used as background (default is terrain.colors(100); unused if col is FALSE)
x	an object of class 'nlsContourRSS'
ask	if TRUE, draw plot interactively (default is FALSE)
useRaster	a bitmap raster is used to plot the image instead of polygons (default is TRUE)
...	further arguments passed to or from other methods

Details

The aim of these functions is to plot the residual sum of squares (RSS) contours which correspond to likelihood contours for a Gaussian model. For each pair of parameters the RSS is calculated on a grid centered on the least squares estimates of both parameters, the other parameters being fixed to their least square estimates. The contours of RSS values are then plotted for each pair of parameters. For each pair of parameters, one of this contour corresponds to a section of the 95 percent Beale's confidence region in the plane of these parameters. This contour is plotted in a different color.

Value

nlsContourRSS returns a list of three objects:

seqPara	a matrix with the sequence of grid values for each parameter
lrss	a list of matrices with logarithm values of RSS in the grid for each pair of parameters
lrss95	the logarithm of the 95 percent residual sum of squares threshold according to Beale (1960)

Author(s)

Florent Baty <florent.baty@gmail.com>
 Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Beale EML (1960) Confidence regions in non-linear estimations. *Journal of the Royal Statistical Society*, **22B**, 41-88.

Bates DM and Watts DG (1988) Nonlinear regression analysis and its applications. Wiley, Chichester, UK.

Examples

```
data(growthcurve4)
nls1 <- nls(gompertz, growthcurve4,
list(lag = 10, mumax = 0.1, LOG10N0 = 6, LOG10Nmax = 9))
crss <- nlsContourRSS(nls1)
plot(crss)
```

nlsJack

Jackknife resampling

Description

Jackknife resampling

Usage

```
nlsJack (nls)
## S3 method for class 'nlsJack'
plot(x, mfr = c(nrow(x$reldif),1), ask = FALSE, ...)
## S3 method for class 'nlsJack'
print(x, ...)
## S3 method for class 'nlsJack'
summary(object, ...)
```

Arguments

nls	an object of class 'nls'
x, object	an object of class 'nlsJack'
mfr	layout definition, default is k rows (k: number of parameters) and 1 column
ask	if TRUE, draw plot interactively
...	further arguments passed to or from other methods

Details

A jackknife resampling procedure is performed. Each observation is sequentially removed from the initial data set using a leave-one-out strategy. A data set with n observations provides thus n resampled data sets of $n-1$ observations. The jackknife estimates with confidence intervals are calculated as described by Seber and Wild (1989) from the results of n new fits of the model on the n jackknife resampled data sets. The leave-one-out procedure is also employed to assess the influence of each observation on each parameter estimate. An observation is empirically defined as influential for one parameter if the difference between the estimate of this parameter with and without the observation exceeds twice the standard error of the estimate divided by \sqrt{n} . This empirical method assumes a small curvature of the nonlinear model. For each parameter, the absolute relative difference (in percent of the estimate) of the estimates with and without each observation is plotted. An asterisk is plotted for each influential observation.

Value

nlsJack returns a list with 7 objects:

estijack	a vector with jackknife estimates
coefjack	a dataframe with the parameter estimates for each jackknife sample
reldif	a dataframe with the absolute relative difference (in percent of the estimate) of the estimates with and without each observation
dfb	a dataframe with dfbetas for each parameter and each observation
jackCI	a dataframe with jackknife confidence intervals
rse	a vector with residual standard error for each jackknife sample
rss	residual a vector with residual sum of squares for each jackknife sample

Author(s)

Florent Baty <florent.baty@gmail.com>

Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Seber GAF, Wild CJ (1989) Nonlinear regression. Wiley, New York.

Examples

```
data(growthcurve4)
nls1 <- nls(gompertz, growthcurve4,
list(lag = 10, mumax = 0.1, LOG10N0 = 6, LOG10Nmax = 9))
lj <- nlsJack(nls1)
plot(lj)
summary(lj)
```

nlsResiduals	<i>NLS residuals</i>
--------------	----------------------

Description

Provides several plots and tests for the analysis of residuals

Usage

```
nlsResiduals (nls)
## S3 method for class 'nlsResiduals'
plot(x, which = 0, ...)
```

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

Arguments

nls	an object of class 'nls'
x	an object of class 'nlsResiduals'
which	an integer: 0 = 4 graphs of residuals (types 1, 2, 4 and 6) 1 = non-transformed residuals against fitted values 2 = standardized residuals against fitted values 3 = sqrt of absolute value of standardized residuals against fitted values 4 = auto-correlation residuals (i+1th residual against ith residual) 5 = histogram of the residuals 6 = qq-plot of the residuals
...	further arguments passed to or from other methods

Details

Several plots and tests are proposed to check the validity of the assumptions of the error model based on the analysis of residuals.

The function `plot.nlsResiduals` proposes several plots of residuals from the nonlinear fit: plot of non-transformed residuals against fitted values, plot of standardized residuals against fitted values, plot of square root of absolute value of standardized residuals against fitted values, auto-correlation plot of residuals (i+1th residual against ith residual), histogram of the non-transformed residuals and normal Q-Q plot of standardized residuals.

`test.nlsResiduals` tests the normality of the residuals with the Shapiro-Wilk test (`shapiro.test` in package `stats`) and the randomness of residuals with the runs test (Siegel and Castellan, 1988). The `runs.test` function used in `nlsTools` is the one implemented in the package `tseries`.

Value

nlsResiduals returns a list of five objects:

std95	the Student value for $\alpha=0.05$ (bilateral) and the degree of freedom of the model
resi1	a matrix with fitted values vs. non-transformed residuals
resi2	a matrix with fitted values vs. standardized residuals
resi3	a matrix with fitted values vs. $\sqrt{\text{abs}(\text{standardized residuals})}$
resi4	a matrix with i th residuals vs. $i+1$ th residuals

Author(s)

Florent Baty <florent.baty@gmail.com>

Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Bates DM and Watts DG (1988) Nonlinear regression analysis and its applications. Wiley, Chichester, UK.

Siegel S and Castellan NJ (1988) Non parametric statistics for behavioral sciences. McGraw-Hill international, New York.

Examples

```
# Plots of residuals
data(growthcurve4)
nls1 <- nls(gompertz, growthcurve4,
list(lag = 10, mumax = 0.1, LOG10N0 = 6, LOG10Nmax = 9))
nr <- nlsResiduals(nls1)
plot(nr, which = 0)

# Histogram and qq-plot
plot(nr, which=5)
plot(nr, which=6)

# Tests
test.nlsResiduals(nr)
```

nlstools

Nonlinear least squares fit

Description

Tools to help the fit of nonlinear models with nls

Usage

```

preview (formula, data, start, variable = 1)
plotfit (x, smooth=FALSE, variable=1, xlab=NULL, ylab=NULL, pch.obs=1, pch.fit="+", lty=1, lwd=1, col.o
overview (x)

```

Arguments

formula	formula of a non-linear model
data	a data frame with header matching the variables given in the formula
start	a list of parameter starting values which names match the parameters given in the formula
variable	index of the variable to be plotted against the predicted values; default is the first independent variable as it appear in the original dataset
x	an object of class 'nls'
smooth	a logical value, default is FALSE. If smooth is TRUE, a plot of observed values is plotted as a function of 1000 values continuously taken in the range interval [min(variable),max(variable)]. This option can only be used if the number of controlled variables is 1.
xlab	X-label
ylab	Y-label
pch.obs	type of point of the observed values
pch.fit	type of point of the fitted values (not applicable if smooth=TRUE)
lty	type of line of the smoothed fitted values (if smooth=TRUE)
lwd	thickness of line of the smoothed fitted values (if smooth=TRUE)
col.obs	color of the observed points
col.fit	color of the fitted values
...	further arguments passed to or from other methods

Details

The function `preview` helps defining the parameter starting values prior fitting the model. It provides a superimposed plot of observed (circles) and predicted (crosses) values of the dependent variable versus one of the independent variables with the model evaluated at the starting values of the parameters. The function `overview` returns the parameters estimates, their standard errors as well as their asymptotic confidence intervals and the correlation matrix (alternately, the function `confint` provides better confidence interval estimates whenever it converges). `plotfit` displays a superimposed plot of the dependent variable versus one the independent variables together with the fitted model.

Author(s)

Florent Baty <florent.baty@gmail.com>
Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Bates DM and Watts DG (1988) Nonlinear regression analysis and its applications. Wiley, Chichester, UK.

See Also

`nls` in the `stats` library and `confint.nls` in the package `MASS`

Examples

```
data(growthcurve4)
preview(gompertz, growthcurve4,
list(lag = 50, mumax = 0.1, LOG10N0 = 6, LOG10Nmax = 8.8))
nls1 <- nls(gompertz, growthcurve4,
list(lag = 10, mumax = 0.1, LOG10N0 = 6, LOG10Nmax = 9))
overview(nls1)
plotfit(nls1, smooth = TRUE)
```

ross

Secondary growth curves

Description

A data frames describing the specific growth rate of *Escherichia coli* as a function of various environmental factors

Usage

```
data(ross)
```

Format

A data frame with five columns (author: the author of the paper from which the data was extracted, T: the temperature in Celsius, aw: the water activity, pH: the pH value, sqrtmumax: the square root of the maximum specific growth rate)

Source

Ross T, Ratkowsky DA, Mellefont LA, McMeekin TA (2003) Modelling the effects of temperature, water activity, pH and lactic acid concentration on the growth rate of *Escherichia coli*. *International of Food Microbiology*, **82**, 33-43.

Examples

```

data(ross)
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2, 2))
plot(ross[c("T", "sqrtmumax")])
plot(ross[c("pH", "sqrtmumax")])
plot(ross[c("aw", "sqrtmumax")])
par(def.par)

```

secondary

*Secondary growth models***Description**

Formulas of secondary growth models commonly used in predictive microbiology

Usage

```

cpm_T
cpm_pH_4p
cpm_pH_3p
cpm_aw_3p
cpm_aw_2p
cpm_T_pH_aw

```

Details

All the models describe the evolution of the square root of the maximum specific growth rate (sqrtmumax) as a function of one or more environmental factors among temperature (T), pH (pH) and water activity (aw). Each model must be fitted to a data frame including at least two columns, the last one named "sqrtmumax" and the first ones named "T", "pH" or "aw" according to the model.

cpm_T is the cardinal temperature model with inflection (Rosso et al., 1993) with four parameters (Tmin, Topt, Tmax, muopt)

cpm_pH_4p is the cardinal pH model (Rosso et al., 1995) with four parameters (pHmin, pHopt, pHmax, muopt)

cpm_pH_3p is a symmetric cardinal pH model with three parameters (pHmin, pHopt, muopt), obtained by fixing pHmax to 2pHopt-pHmin in the cpm_pH_4p model

cpm_aw_3p is the cardinal aw model (Rosso and Robinson, 2001) with three parameters (awmin, awopt, muopt)

cpm_aw_2p is a simplified cardinal aw model (Rosso and Robinson, 2001) with two parameters (awmin, muopt) obtained by fixing awopt to 1 in the cpm_aw_3p model

cpm_T_pH_aw is the cardinal model based on the gamma concept (Pinon et al., 2004) with 9 parameters (Tmin, Topt, Tmax, pHmin, pHopt, pHmax, awmin, awopt, muopt)

Value

A formula

Author(s)

Florent Baty <florent.baty@gmail.com>

Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Pinon A, Zwietering M, Perrier L, Membrève J, Leporq B, Mettler E, Thuault D, Coroller L, Stahl V, Vialette M (2004) Development and validation of experimental protocols for use of cardinal models for prediction of microorganism growth in food products. *Applied Environmental Microbiology*, **70**, 1081-1087.

Rosso L, Robinson TP (2001) A cardinal model to describe the effect of water activity on the growth of moulds. *International Journal of Food Microbiology*, **63**, 265-273.

Rosso L, Lobry JR, Bajard S, Flandrois JP (1995) Convenient model to describe the combined effects of temperature and pH on microbial growth. *Applied Environmental Microbiology*, **61**, 610-616.

Rosso L, Lobry JR, Flandrois JP (1993) An unexpected correlation between cardinal temperatures of microbial growth highlighted by a new model. *Journal of Theoretical Biology* **162**, 447-463.

Examples

```
data(ross)

# Example for the cpm_T model

d1 <- subset(ross, author == "salter" & aw == 0.997,
select = c(T, sqrtmumax))
nls1 <- nls(cpm_T, d1, list(muopt = 1.7, Tmin = 4, Topt = 40, Tmax = 47))
plotfit(nls1, smooth = TRUE)
overview(nls1)

# Example for the cpm_pH_4p model

d2 <- subset(ross, author == "presser" & aw > 0.99,
select = c(pH, sqrtmumax))
nls2 <- nls(cpm_pH_4p, d2, list(muopt = 0.5, pHmin = 4,
pHopt = 6.5, pHmax = 9))
plotfit(nls2, smooth = TRUE)
```

```

overview(nls2)

# Example for the cpm_pH_3p model

d3 <- subset(ross, author == "presser" & aw == 0.997,
select = c(pH, sqrtmumax))
nls3 <- nls(cpm_pH_3p, d3, list(muopt = 0.5, pHmin = 4, pHopt = 6.5))
plotfit(nls3, smooth = TRUE)
overview(nls3)

# Example for the cpm_aw_3p model

d4<-subset(ross, author == "mellefont", select = c(aw, sqrtmumax))
nls4 <- nls(cpm_aw_3p, d4, list(muopt = 0.6, awmin = 0.95, awopt = 0.995))
plotfit(nls4, smooth = TRUE)
overview(nls4)

# Example for the cpm_aw_2p model

d5 <- subset(ross, author == "mellefont" & aw < 0.99,
select = c(aw, sqrtmumax))
nls5 <- nls(cpm_aw_2p, d5, list(muopt = 0.6, awmin = 0.95))
plotfit(nls5, smooth = TRUE)
overview(nls5)

# Examples for the cpm_T_pH_aw model

d6 <- subset(ross, select = c(T, pH, aw, sqrtmumax))
nls6 <- nls(cpm_T_pH_aw, d6, list(muopt = 2, Tmin = 4, Topt = 40, Tmax = 49,
pHmin = 4, pHopt = 6.5, pHmax = 9, awmin = 0.95, awopt = 0.995))
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2, 2))
plotfit(nls6, variable = 1)
plotfit(nls6, variable = 2)
plotfit(nls6, variable = 3)
overview(nls6)
par(def.par)

```

survivalcurve

Bacterial survival data sets

Description

Bacterial kinetics data sets

Usage

```

data(survivalcurve1)
data(survivalcurve2)
data(survivalcurve3)

```

Format

Data frames with 2 columns (t: time, LOG10N: decimal logarithm of bacterial density)

Source

Data obtained by Florent Baty <florent.baty@kssg.ch> and Marie-Laure Delignette-Muller <ml.delignette@vet-lyon>

Examples

```
data(survivalcurve1)
data(survivalcurve2)
data(survivalcurve3)
def.par <- par(no.readonly = TRUE)
par(mfrow=c(2,2))
plot(survivalcurve1, type="b")
plot(survivalcurve2, type="b")
plot(survivalcurve3, type="b")
par(def.par)
```

survivalmodels

Bacterial survival models

Description

Formulas of primary survival models commonly used in predictive microbiology

Usage

```
geeraerd
geeraerd_without_Nres
geeraerd_without_Sl
mafart
albert
trilinear
bilinear_without_Nres
bilinear_without_Sl
```

Details

These models describe the evolution of the decimal logarithm of the microbial count (LOG10N) as a function of the time (t).

geeraerd is the model of Geeraerd et al. (2005) with four parameters (LOG10N0, kmax, Sl, LOG10Nres)

geeraerd_without_Nres is the model of Geeraerd et al. (2005) with three parameters (LOG10N0, kmax, Sl), without tail

geeraerd_without_Sl is the model of Geeraerd et al. (2005) with three parameters (LOG10N0,

kmax, Nres), without shoulder

mafart is the Weibull model as parameterized by Mafart et al. (2002) with three parameters (p, delta, LOG10N0)

albert is the modified Weibull model proposed by Albert and Mafart (2005) with four parameters (p, delta, LOG10N0, LOG10Nres)

trilinear is the three-phase linear model with four parameters (LOG10N0, kmax, S1, LOG10Nres)

bilinear_without_Nres is the two-phase linear model with three parameters (LOG10N0, kmax, S1), without tail

bilinear_without_S1 is the two-phase linear model with three parameters (LOG10N0, kmax, LOG10Nres), without shoulder

Value

A formula

Author(s)

Florent Baty <florent.baty@gmail.com>

Marie-Laure Delignette-Muller <ml.delignette@vetagro-sup.fr>

References

Albert I, Mafart P (2005) A modified Weibull model for bacterial inactivation. *International Journal of Food Microbiology*, **100**, 197-211.

Geeraerd AH, Valdramidis VP, Van Impe JF (2005) GInaFiT, a freeware tool to assess non-log-linear microbial survivor curves. *International Journal of Food Microbiology*, **102**, 95-105.

Mafart P, Couvert O, Gaillard S, Leguerinel I (2002) On calculating sterility in thermal preservation methods : application of the Weibull frequency distribution model. *International Journal of Food Microbiology*, **72**, 107-113.

Examples

```
# Example 1

data(survivalcurve1)
nls1a <- nls(geeraerd, survivalcurve1,
list(S1 = 5, kmax = 1.5, LOG10N0 = 7, LOG10Nres = 1))
nls1b <- nls(trilinear, survivalcurve1,
list(S1 = 5, kmax = 1.5, LOG10N0 = 7, LOG10Nres = 1))
nls1c <- nls(albert, survivalcurve1,
list(p = 1.2, delta = 4, LOG10N0 = 7, LOG10Nres = 1))
def.par <- par(no.readonly = TRUE)
```

```
par(mfrow = c(2,2))
overview(nls1a)
plotfit(nls1a, smooth = TRUE)
overview(nls1b)
plotfit(nls1b, smooth = TRUE)
overview(nls1c)
plotfit(nls1c, smooth = TRUE)
par(def.par)

# Example 2

data(survivalcurve2)
nls2a <- nls(geeraerd_without_Nres, survivalcurve2,
list(S1 = 10, kmax = 1.5, LOG10N0 = 7.5))
nls2b <- nls(bilinear_without_Nres, survivalcurve2,
list(S1 = 10, kmax = 1.5, LOG10N0 = 7.5))
nls2c <- nls(mafart, survivalcurve2,
list(p = 1.5, delta = 8, LOG10N0 = 7.5))
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2,2))
overview(nls2a)
plotfit(nls2a, smooth = TRUE)
overview(nls2b)
plotfit(nls2b, smooth = TRUE)
overview(nls2c)
plotfit(nls2c, smooth = TRUE)
par(def.par)

# Example 3

data(survivalcurve3)
nls3a <- nls(geeraerd_without_S1, survivalcurve3,
list(kmax = 4, LOG10N0 = 7.5, LOG10Nres = 1))
nls3b <- nls(bilinear_without_S1, survivalcurve3,
list(kmax = 4, LOG10N0 = 7.5, LOG10Nres = 1))
nls3c <- nls(mafart, survivalcurve3,
list(p = 0.5, delta = 0.2, LOG10N0 = 7.5))
def.par <- par(no.readonly = TRUE)
par(mfrow = c(2,2))
overview(nls3a)
plotfit(nls3a, smooth = TRUE)
overview(nls3b)
plotfit(nls3b, smooth = TRUE)
overview(nls3c)
plotfit(nls3c, smooth = TRUE)
par(def.par)
```

Index

*Topic **datasets**

- competitioncurve, 2
- growthcurve, 5
- michaelisdata, 8
- ross, 20
- survivalcurve, 23

*Topic **models**

- competitionmodels, 3
- growthmodels, 6
- michaelismodels, 9
- secondary, 21
- survivalmodels, 24

*Topic **nonlinear**

- nlsBoot, 10
- nlsConfRegions, 12
- nlsContourRSS, 14
- nlsJack, 15
- nlsResiduals, 17
- nlstools, 18

albert (survivalmodels), 24

baranyi (growthmodels), 6
baranyi_without_lag (growthmodels), 6
baranyi_without_Nmax (growthmodels), 6
bilinear_without_Nres (survivalmodels), 24
bilinear_without_Sl (survivalmodels), 24
buchanan (growthmodels), 6
buchanan_without_lag (growthmodels), 6
buchanan_without_Nmax (growthmodels), 6

compet_mich (michaelismodels), 9
competition1 (competitioncurve), 2
competition2 (competitioncurve), 2
competitioncurve, 2
competitionmodels, 3
cpm_aw_2p (secondary), 21
cpm_aw_3p (secondary), 21
cpm_pH_3p (secondary), 21

cpm_pH_4p (secondary), 21
cpm_T (secondary), 21
cpm_T_pH_aw (secondary), 21

geeraerd (survivalmodels), 24
geeraerd_without_Nres (survivalmodels), 24
geeraerd_without_Sl (survivalmodels), 24
gompertz (growthmodels), 6
growthcurve, 5
growthcurve1 (growthcurve), 5
growthcurve2 (growthcurve), 5
growthcurve3 (growthcurve), 5
growthcurve4 (growthcurve), 5
growthmodels, 6

jameson_baranyi (competitionmodels), 3
jameson_buchanan (competitionmodels), 3
jameson_without_lag (competitionmodels), 3

mafart (survivalmodels), 24
michaelis (michaelismodels), 9
michaelisdata, 8
michaelismodels, 9

nlsBoot, 10
nlsConfRegions, 12
nlsContourRSS, 14
nlsJack, 15
nlsResiduals, 17
nlstools, 18
non_compet_mich (michaelismodels), 9

overview (nlstools), 18

plot.nlsBoot (nlsBoot), 10
plot.nlsConfRegions (nlsConfRegions), 12
plot.nlsContourRSS (nlsContourRSS), 14
plot.nlsJack (nlsJack), 15
plot.nlsResiduals (nlsResiduals), 17

plotfit (nlstools), 18
preview (nlstools), 18
print.nlsBoot (nlsBoot), 10
print.nlsConfRegions (nlsConfRegions),
12
print.nlsContourRSS (nlsContourRSS), 14
print.nlsJack (nlsJack), 15
print.nlsResiduals (nlsResiduals), 17

ross, 20

secondary, 21
summary.nlsBoot (nlsBoot), 10
summary.nlsJack (nlsJack), 15
survivalcurve, 23
survivalcurve1 (survivalcurve), 23
survivalcurve2 (survivalcurve), 23
survivalcurve3 (survivalcurve), 23
survivalmodels, 24

test.nlsResiduals (nlsResiduals), 17
trilinear (survivalmodels), 24

vmkm (michaelisdata), 8
vmkmi (michaelisdata), 8