Package 'drda'

January 16, 2025

Type Package

Title Dose-Response Data Analysis

Version 2.0.5

Description Fit logistic functions to observed dose-response continuous data and evaluate goodness-of-fit measures. See Malyutina A., Tang J., and Pessia A. (2023) <doi:10.18637/jss.v106.i04>.

License MIT + file LICENSE

URL https://github.com/albertopessia/drda

BugReports https://github.com/albertopessia/drda/issues

Depends R (>= 3.6.0)

Imports graphics, grDevices, stats

Suggests knitr, rmarkdown, spelling, testthat (>= 3.1.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Alberto Pessia [aut, cre] (<https://orcid.org/0000-0001-8607-9191>), Alina Malyutina [ctb] (<https://orcid.org/0000-0001-8220-5859>)

Maintainer Alberto Pessia <dev@albertopessia.com>

Repository CRAN

Date/Publication 2025-01-16 10:00:02 UTC

Contents

drda — *Fit non-linear growth curves*

Description

Use the Newton's with a trust-region method to fit non-linear growth curves to observed data.

Usage

```
drda(
  formula,
  data,
  subset,
  weights,
  na.action,
  mean_function = "logistic4",
  lower_bound = NULL,
  upper_bound = NULL,
  start = NULL,
  max_iter = 1000
)
```

Arguments

`formula`	an object of class `formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted. Currently supports only formulas of the type `y ~ x`.
`data`	an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `drda` is called.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.
`weights`	an optional vector of weights to be used in the fitting process. If provided, weighted least squares is used with weights `weights` (that is, minimizing `sum(weights * residuals^2)`), otherwise ordinary least squares is used.
`na.action`	a function which indicates what should happen when the data contain `NA`s. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The 'factory-fresh' default is `na.omit`. Another possible value is `NULL`, no action. Value `na.exclude` can be useful.
`mean_function`	the model to be fitted. See `details` for available models.
`lower_bound`	numeric vector with the minimum admissible values of the parameters. Use `-Inf` to specify an unbounded parameter.
`upper_bound`	numeric vector with the maximum admissible values of the parameters. Use `Inf` to specify an unbounded parameter.

start starting values for the parameters.

max_iter maximum number of iterations in the optimization algorithm.

Details

Available models:

Generalized (5-parameter) logistic function:
The 5-parameter logistic function can be selected by choosing mean_function = "logistic5" or mean_function = "l5". The function is defined here as
alpha + delta / (1 + nu * exp(-eta * (x - phi)))^(1 / nu)
where eta > 0 and nu > 0. When delta is positive (negative) the curve is monotonically increasing (decreasing).
Parameter alpha is the value of the function when x -> -Inf. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi is related to the mid-value of the function. Parameter nu affects near which asymptote maximum growth occurs.
The value of the function when x -> Inf is alpha + delta. In dose-response studies delta can be interpreted as the maximum theoretical achievable effect.

4-parameter logistic function:
The 4-parameter logistic function is the default model of drda. It can be explicitly selected by choosing mean_function = "logistic4" or mean_function = "l4". The function is obtained by setting nu = 1 in the generalized logistic function, that is
alpha + delta / (1 + exp(-eta * (x - phi)))
where eta > 0. When delta is positive (negative) the curve is monotonically increasing (decreasing).
Parameter alpha is the value of the function when x -> -Inf. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi represents the x value at which the curve is equal to its mid-point, i.e. f(phi; alpha, delta, eta, phi) = alpha +
The value of the function when x -> Inf is alpha + delta. In dose-response studies delta can be interpreted as the maximum theoretical achievable effect.

2-parameter logistic function:
The 2-parameter logistic function can be selected by choosing mean_function = "logistic2" or mean_function = "l2". For a monotonically increasing curve set nu = 1, alpha = 0, and delta = 1:
1 / (1 + exp(-eta * (x - phi)))
For a monotonically decreasing curve set nu = 1, alpha = 1, and delta = -1:
1 - 1 / (1 + exp(-eta * (x - phi)))
where eta > 0. The lower bound of the curve is zero while the upper bound of the curve is one. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi represents the x value at which the curve is equal to its mid-point, i.e. f(phi; eta, phi) = 1 / 2.

Gompertz function:
The Gompertz function is the limit for nu -> 0 of the 5-parameter logistic function. It can be selected by choosing mean_function = "gompertz" or mean_function = "gz". The function is defined in this package as
alpha + delta * exp(-exp(-eta * (x - phi)))
where eta > 0.

Parameter alpha is the value of the function when x -> -Inf. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi sets the displacement along the x-axis.
The value of the function when x -> Inf is alpha + delta. In dose-response studies delta can be interpreted as the maximum theoretical achievable effect.
The mid-point of the function, that is alpha + delta / 2, is achieved at x = phi - log(log(2)) / eta.

Generalized (5-parameter) log-logistic function:
The 5-parameter log-logistic function is selected by setting mean_function = "loglogistic5" or mean_function = "ll5". The function is defined here as
alpha + delta * (x^eta / (x^eta + nu * phi^eta))^(1 / nu)
where x >= 0, eta > 0, phi > 0, and nu > 0. When delta is positive (negative) the curve is monotonically increasing (decreasing). The function is defined only for positive values of the predictor variable x.
Parameter alpha is the value of the function at x = 0. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi is related to the mid-value of the function. Parameter nu affects near which asymptote maximum growth occurs.
The value of the function when x -> Inf is alpha + delta. In dose-response studies delta can be interpreted as the maximum theoretical achievable effect.

4-parameter log-logistic function:
The 4-parameter log-logistic function is selected by setting mean_function = "loglogistic4" or mean_function = "ll4". The function is obtained by setting nu = 1 in the generalized log-logistic function, that is
alpha + delta * x^eta / (x^eta + phi^eta)
where x >= 0 and eta > 0. When delta is positive (negative) the curve is monotonically increasing (decreasing). The function is defined only for positive values of the predictor variable x.
Parameter alpha is the value of the function at x = 0. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi represents the x value at which the curve is equal to its mid-point, i.e. f(phi; alpha, delta, eta, phi) = alpha + delta
The value of the function when x -> Inf is alpha + delta. In dose-response studies delta can be interpreted as the maximum theoretical achievable effect.

2-parameter log-logistic function:
The 2-parameter log-logistic function is selected by setting mean_function = "loglogistic2" or mean_function = "ll2". For a monotonically increasing curve set nu = 1, alpha = 0, and delta = 1:
x^eta / (x^eta + phi^eta)
For a monotonically decreasing curve set nu = 1, alpha = 1, and delta = -1:
1 - x^eta / (x^eta + phi^eta)
where x >= 0, eta > 0, and phi > 0. The lower bound of the curve is zero while the upper bound of the curve is one.
Parameter eta represents the steepness (growth rate) of the curve. Parameter phi represents the x value at which the curve is equal to its mid-point, i.e. f(phi; eta, phi) = 1 / 2.

log-Gompertz function:
The log-Gompertz function is the limit for nu -> 0 of the 5-parameter log-logistic function. It can be selected by choosing mean_function = "loggompertz" or mean_function = "lgz". The function is defined in this package as

alpha + delta * exp(-(phi / x)^eta)
where x > 0, eta > 0, and phi > 0. Note that the limit for x -> 0 is alpha. When delta is positive (negative) the curve is monotonically increasing (decreasing). The function is defined only for positive values of the predictor variable x.
Parameter alpha is the value of the function at x = 0. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi sets the displacement along the x-axis.
The value of the function when x -> Inf is alpha + delta. In dose-response studies delta can be interpreted as the maximum theoretical achievable effect.

Constrained optimization:
It is possible to search for the maximum likelihood estimates within pre-specified interval regions.
Note: Hypothesis testing is not available for constrained estimates because asymptotic approximations might not be valid.

Value

An object of class drda and model_fit, where model is the chosen mean function. It is a list containing the following components:

converged boolean value assessing if the optimization algorithm converged or not.

iterations total number of iterations performed by the optimization algorithm

constrained boolean value set to TRUE if optimization was constrained.

estimated boolean vector indicating which parameters were estimated from the data.

coefficients maximum likelihood estimates of the model parameters.

rss minimum value (found) of the residual sum of squares.

df.residuals residual degrees of freedom.

fitted.values fitted mean values.

residuals residuals, that is response minus fitted values.

weights (only for weighted fits) the specified weights.

mean_function model that was used for fitting.

n effective sample size.

sigma corrected maximum likelihood estimate of the standard deviation.

loglik maximum value (found) of the log-likelihood function.

fisher.info observed Fisher information matrix evaluated at the maximum likelihood estimator.

vcov approximate variance-covariance matrix of the model parameters.

call the matched call.

terms the terms object used.

model the model frame used.

na.action (where relevant) information returned by model.frame on the special handling of NAs.

Examples

```
# by default `drda` uses a 4-parameter logistic function for model fitting
fit_l4 <- drda(response ~ log_dose, data = voropm2)

# get a general overview of the results
summary(fit_l4)

# compare the model against a flat horizontal line and the full model
anova(fit_l4)

# 5-parameter logistic curve appears to be a better model
fit_l5 <- drda(response ~ log_dose, data = voropm2, mean_function = "l5")
plot(fit_l4, fit_l5)

# fit a 2-parameter logistic function
fit_l2 <- drda(response ~ log_dose, data = voropm2, mean_function = "l2")

# compare our models
anova(fit_l2, fit_l4)

# use log-logistic functions when utilizing doses (instead of log-doses)
# here we show the use of other arguments as well
fit_ll5 <- drda(
  response ~ dose, weights = weight, data = voropm2,
  mean_function = "loglogistic5", lower_bound = c(0.5, -1.5, 0, -Inf, 0.25),
  upper_bound = c(1.5, 0.5, 5, Inf, 3), start = c(1, -1, 3, 100, 1),
  max_iter = 10000
)

# note that the maximum likelihood estimate is outside the region of
# optimization: not only the variance-covariance matrix is now singular but
# asymptotic assumptions do not hold anymore.
```

---

effective_dose *Effective dose*

Description

Estimate effective doses, that is the x values for which f(x) = y.

Usage

```
effective_dose(object, y, type, level)
```

Arguments

object	fit object as returned by drda.
y	numeric vector with response levels (default 0.5).
type	character string with either "relative" (default) or "absolute".
level	level of confidence intervals (default 0.95).

Details

Given a fitted model f(x; theta) we seek the values x at which the function is equal to the specified response values.

If responses are given on a relative scale (type = "relative"), then y is expected to range in the interval (0, 1).

If responses are given on an absolute scale (type = "absolute"), then y is free to vary on the whole real line. Note, however, that the solution does not exist when y is not in the image of the function.

Value

Numeric matrix with the effective doses and the corresponding confidence intervals. Each row is associated with each value of y.

Examples

```
drda_fit <- drda(response ~ log_dose, data = voropm2)
effective_dose(drda_fit)

# relative values are given on the (0, 1) range
effective_dose(drda_fit, y = c(0.2, 0.8))

# explicitly say when we are using actual response values
effective_dose(drda_fit, y = c(0.2, 0.8), type = "absolute")

# use a different confidence level
effective_dose(drda_fit, y = 0.6, level = 0.8)
```

---

gompertz_fn *Gompertz function*

Description

Evaluate at a particular set of parameters the Gompertz function.

Usage

```
gompertz_fn(x, theta)
```

Arguments

x	numeric vector at which the Gompertz function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The Gompertz function `f(x; theta)` is defined here as
`g(x; theta) = exp(-exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi)`, `alpha` is the value of the function when `x -> -Inf`, `delta` is the (signed) height of the curve, `eta > 0` is the steepness of the curve or growth rate, and `phi` is related with the value of function at `x = 0`.

When `delta < 0` the curve is monotonically decreasing while it is monotonically increasing for `delta > 0`.

Value

Numeric vector of the same length of `x` with the values of the Gompertz function.

`gompertz_gradient` *Gompertz function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the Gompertz function.

Usage

```
gompertz_gradient(x, theta)

gompertz_hessian(x, theta)

gompertz_gradient_hessian(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the four parameters in the form `c(alpha, delta, eta, phi)`.

Details

The Gompertz function `f(x; theta)` is defined here as
`g(x; theta) = exp(-exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi)` and `eta > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

Value

Gradient or Hessian evaluated at the specified point.

gompertz_gradient_2 *Gompertz function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the Gompertz function.

Usage

```
gompertz_gradient_2(x, theta)

gompertz_hessian_2(x, theta)

gompertz_gradient_hessian_2(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The Gompertz function f(x; theta) is defined here as
g(x; theta) = exp(-exp(-eta * (x - phi))) f(x; theta) = alpha + delta g(x; theta)

where theta = c(alpha, delta, eta, phi) and eta > 0. When delta is positive (negative) the curve is monotonically increasing (decreasing).

This set of functions use a different parameterization from link[drda]{gompertz_gradient}. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with eta2 = log(eta).

Note that argument theta is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

loggompertz_fn *log-Gompertz function*

Description

Evaluate at a particular set of parameters the log-Gompertz function.

Usage

```
loggompertz_fn(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The log-Gompertz function f(x; theta) is defined here as
f(x; theta) = alpha + delta exp(-(phi / x)^eta)

where x >= 0, theta = c(alpha, delta, eta, phi), eta > 0, and phi > 0. By convention we set f(0; theta) = lim_{x -> 0} f(x; theta) = alpha.

Value

Numeric vector of the same length of x with the values of the log-logistic function.

loggompertz_gradient *Log-Gompertz function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the log-Gompertz function.

Usage

```
loggompertz_gradient(x, theta)

loggompertz_hessian(x, theta)

loggompertz_gradient_hessian(x, theta)
```

Arguments

x — numeric vector at which the function is to be evaluated.

theta — numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The log-Gompertz function f(x; theta) is defined here as
f(x; theta) = alpha + delta exp(-(phi / x)^eta)

where x >= 0, theta = c(alpha, delta, eta, phi), eta > 0, and phi > 0. By convention we set f(0; theta) = lim_{x -> 0} f(x; theta) = alpha.

Value

Gradient or Hessian evaluated at the specified point.

---

loggompertz_gradient_2 *Log-Gompertz function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the log-Gompertz function.

Usage

```
loggompertz_gradient_2(x, theta)

loggompertz_hessian_2(x, theta)

loggompertz_gradient_hessian_2(x, theta)
```

Arguments

x — numeric vector at which the function is to be evaluated.

theta — numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The log-Gompertz function f(x; theta) is defined here as
f(x; theta) = alpha + delta exp(-(phi / x)^eta)

where x >= 0, theta = c(alpha, delta, eta, phi), eta > 0, and phi > 0. By convention we set f(0; theta) = lim_{x -> 0} f(x; theta) = alpha.

This set of functions use a different parameterization from `link[drda]{loggompertz_gradient}`. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with `eta2` = `log(eta)` and `phi2` = `log(phi)`.

Note that argument `theta` is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

---

logistic2_fn — *2-parameter logistic function*

Description

Evaluate at a particular set of parameters the 2-parameter logistic function.

Usage

```
logistic2_fn(x, theta)
```

Arguments

`x`	numeric vector at which the logistic function is to be evaluated.
`theta`	numeric vector with the four parameters in the form `c(alpha, delta, eta, phi)`. `alpha` can only be equal to 0 or 1 while `delta` can only be equal to 1 or -1.

Details

The 2-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta` = `c(alpha, delta, eta, phi)` and `eta` > `0`. Only `eta` and `phi` are free to vary (therefore the name) while vector `c(alpha, delta)` is constrained to be either `c(0, 1)` (monotonically increasing curve) or `c(1, -1)` (monotonically decreasing curve).

This function allows values other than `{0, 1, -1}` for `alpha` and `delta` but will coerce them to their proper constraints.

Value

Numeric vector of the same length of `x` with the values of the logistic function.

logistic2_gradient — *2-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 2-parameter logistic function.

Usage

```
logistic2_gradient(x, theta, delta)

logistic2_hessian(x, theta, delta)

logistic2_gradient_hessian(x, theta, delta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the two parameters in the form `c(eta, phi)`.
`delta`	value of delta parameter (either 1 or -1).

Details

The 2-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi)` and `eta > 0`. Only `eta` and `phi` are free to vary (therefore the name) while vector `c(alpha, delta)` is constrained to be either `c(0, 1)` (monotonically increasing curve) or `c(1, -1)` (monotonically decreasing curve).

Value

Gradient or Hessian evaluated at the specified point.

logistic2_gradient_2 — *2-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 2-parameter logistic function.

Usage

```
logistic2_gradient_2(x, theta, delta)

logistic2_hessian_2(x, theta, delta)

logistic2_gradient_hessian_2(x, theta, delta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the two parameters in the form `c(eta, phi)`.
`delta`	value of delta parameter (either 1 or -1).

Details

The 2-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi)` and `eta > 0`. Only `eta` and `phi` are free to vary (therefore the name) while vector `c(alpha, delta)` is constrained to be either `c(0, 1)` (monotonically increasing curve) or `c(1, -1)` (monotonically decreasing curve).

This set of functions use a different parameterization from `link[drda]{logistic2_gradient}`. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with `eta2 = log(eta)`.

Note that argument `theta` is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

`logistic4_fn` *4-parameter logistic function*

Description

Evaluate at a particular set of parameters the 4-parameter logistic function.

Usage

```
logistic4_fn(x, theta)
```

Arguments

`x`	numeric vector at which the logistic function is to be evaluated.
`theta`	numeric vector with the four parameters in the form `c(alpha, delta, eta, phi)`.

Details

The 4-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi)`, `alpha` is the value of the function when `x -> -Inf`, `delta` is the (signed) height of the curve, `eta > 0` is the steepness of the curve or growth rate (also known as the Hill coefficient), and `phi` is the value of `x` at which the curve is equal to its mid-point.

When `delta < 0` the curve is monotonically decreasing while it is monotonically increasing for `delta > 0`.

The mid-point `f(phi; theta)` is equal to `alpha + delta / 2` while the value of the function for `x -> Inf` is `alpha + delta`.

Value

Numeric vector of the same length of `x` with the values of the logistic function.

---

logistic4_gradient — *4-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 4-parameter logistic function.

Usage

```
logistic4_gradient(x, theta)

logistic4_hessian(x, theta)

logistic4_gradient_hessian(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the four parameters in the form `c(alpha, delta, eta, phi)`.

Details

The 4-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi)` and `eta > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

Value

Gradient or Hessian evaluated at the specified point.

logistic4_gradient_2 *4-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 4-parameter logistic function.

Usage

```
logistic4_gradient_2(x, theta)

logistic4_hessian_2(x, theta)

logistic4_gradient_hessian_2(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the four parameters in the form `c(alpha, delta, eta, phi)`.

Details

The 4-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + exp(-eta * (x - phi)))` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi)` and `eta > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

This set of functions use a different parameterization from `link[drda]{logistic4_gradient}`. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with `eta2 = log(eta)`.

Note that argument `theta` is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

logistic5_fn *5-parameter logistic function*

Description

Evaluate at a particular set of parameters the 5-parameter logistic function.

Usage

```
logistic5_fn(x, theta)
```

Arguments

x	numeric vector at which the logistic function is to be evaluated.
theta	numeric vector with the five parameters in the form c(alpha, delta, eta, phi, nu).

Details

The 5-parameter logistic function f(x; theta) is defined here as
g(x; theta) = 1 / (1 + nu * exp(-eta * (x - phi)))^(1 / nu) f(x; theta) = alpha + delta g(x; theta)

where theta = c(alpha, delta, eta, phi, nu), eta > 0, and nu > 0.

When delta is positive (negative) the curve is monotonically increasing (decreasing). When x -> -Inf the value of the function is alpha while the value of the function for x -> Inf is alpha + delta .

Parameter alpha is the value of the function when x -> -Inf. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi is related to the mid-value of the function. Parameter nu affects near which asymptote maximum growth occurs.

Value

Numeric vector of the same length of x with the values of the logistic function.

logistic5_gradient *5-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 5-parameter logistic function.

Usage

```
logistic5_gradient(x, theta)

logistic5_hessian(x, theta)

logistic5_gradient_hessian(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the five parameters in the form `c(alpha, delta, eta, phi, nu)`.

Details

The 5-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + nu * exp(-eta * (x - phi)))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi, nu)`, `eta > 0`, and `nu > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

Value

Gradient or Hessian evaluated at the specified point.

`logistic5_gradient_2` *5-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 5-parameter logistic function.

Usage

```
logistic5_gradient_2(x, theta)

logistic5_hessian_2(x, theta)

logistic5_gradient_hessian_2(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the five parameters in the form `c(alpha, delta, eta, phi, nu)`.

Details

The 5-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (1 + nu * exp(-eta * (x - phi)))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi, nu)`, `eta > 0`, and `nu > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

This set of functions use a different parameterization from `link[drda]{logistic5_gradient}`. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with `eta2 = log(eta)` and `nu2 = log(nu)`.

Note that argument `theta` is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

---

logistic6_fn — *6-parameter logistic function*

Description

Evaluate at a particular set of parameters the 6-parameter logistic function.

Usage

```
logistic6_fn(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the six parameters in the form `c(alpha, delta, eta, phi, nu, xi)`.

Details

The 6-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (xi + nu * exp(-eta * (x - phi)))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi, nu, xi)`, `eta > 0`, `nu > 0`, and `xi > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

Parameter `alpha` is the value of the function when `x -> -Inf`. Parameter `delta` affects the value of the function when `x -> Inf`. Parameter `eta` represents the steepness (growth rate) of the curve. Parameter `phi` is related to the mid-value of the function. Parameter `nu` affects near which asymptote maximum growth occurs. Parameter `xi` affects the value of the function when `x -> Inf`.

Note: The 6-parameter logistic function is over-parameterized and non-identifiable from data. It is available only for theoretical research.

Value

Numeric vector of the same length of `x` with the values of the logistic function.

logistic6_gradient *6-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 6-parameter logistic function.

Usage

```
logistic6_gradient(x, theta)

logistic6_hessian(x, theta)

logistic6_gradient_hessian(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the six parameters in the form `c(alpha, delta, eta, phi, nu, xi)`.

Details

The 6-parameter logistic function `f(x; theta)` is defined here as
`g(x; theta) = 1 / (xi + nu * exp(-eta * (x - phi)))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `theta = c(alpha, delta, eta, phi, nu, xi)`, `eta > 0`, `nu > 0`, and `xi > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

Note: The 6-parameter logistic function is over-parameterized and non-identifiable from data. It is available only for theoretical research.

Value

Gradient or Hessian evaluated at the specified point.

logistic6_gradient_2 *6-parameter logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 6-parameter logistic function.

Usage

```
logistic6_gradient_2(x, theta)

logistic6_hessian_2(x, theta)

logistic6_gradient_hessian_2(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the six parameters in the form c(alpha, delta, eta, phi, nu, xi).

Details

The 6-parameter logistic function f(x; theta) is defined here as
g(x; theta) = 1 / (xi + nu * exp(-eta * (x - phi)))^(1 / nu) f(x; theta) = alpha + delta g(x; theta)

where theta = c(alpha, delta, eta, phi, nu, xi), eta > 0, nu > 0, and xi > 0. When delta is positive (negative) the curve is monotonically increasing (decreasing).

This set of functions use a different parameterization from link[drda]{logistic6_gradient}. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with eta2 = log(eta), nu2 = log(nu), and xi2 = log(xi).

Note that argument theta is on the original scale and not on the log scale.

Note: The 6-parameter logistic function is over-parameterized and non-identifiable from data. It is available only for theoretical research.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

---

loglogistic2_fn *2-parameter log-logistic function*

Description

Evaluate at a particular set of parameters the 2-parameter log-logistic function.

Usage

```
loglogistic2_fn(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha, delta, eta, phi). alpha can only be equal to 0 or 1 while delta can only be equal to 1 or -1.

Details

The 2-parameter log-logistic function `f(x; theta)` is defined here as
`g(x; theta) = x^eta / (x^eta + phi^eta)` `f(x; theta) = alpha + delta g(x; theta)`

where `x >= 0`, `theta = c(alpha, delta, eta, phi)`, `eta > 0`, and `phi > 0`. Only `eta` and `phi` are free to vary (therefore the name) while vector `c(alpha, delta)` is constrained to be either `c(0, 1)` (monotonically increasing curve) or `c(1, -1)` (monotonically decreasing curve).

This function allows values other than `{0, 1, -1}` for `alpha` and `delta` but will coerce them to their proper constraints.

Value

Numeric vector of the same length of `x` with the values of the log-logistic function.

---

loglogistic2_gradient *2-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 2-parameter log-logistic function.

Usage

```
loglogistic2_gradient(x, theta, delta)

loglogistic2_hessian(x, theta, delta)

loglogistic2_gradient_hessian(x, theta, delta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the two parameters in the form `c(eta, phi)`.
`delta`	value of delta parameter (either 1 or -1).

Details

The 2-parameter log-logistic function `f(x; theta)` is defined here as
`g(x; theta) = x^eta / (x^eta + phi^eta)` `f(x; theta) = alpha + delta g(x; theta)`

where `x >= 0`, `theta = c(alpha, delta, eta, phi)`, `eta > 0`, and `phi > 0`. Only `eta` and `phi` are free to vary (therefore the name), while `c(alpha, delta)` are constrained to be either `c(0, 1)` (monotonically increasing curve) or `c(1, -1)` (monotonically decreasing curve).

Value

Gradient or Hessian evaluated at the specified point.

loglogistic2_gradient_2 *2-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 2-parameter log-logistic function.

Usage

```
loglogistic2_gradient_2(x, theta, delta)

loglogistic2_hessian_2(x, theta, delta)

loglogistic2_gradient_hessian_2(x, theta, delta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the two parameters in the form c(eta, phi).
delta	value of delta parameter (either 1 or -1).

Details

The 2-parameter log-logistic function f(x; theta) is defined here as
g(x; theta) = x^eta / (x^eta + phi^eta) f(x; theta) = alpha + delta g(x; theta)

where x >= 0, theta = c(alpha, delta, eta, phi), eta > 0, and phi > 0. Only eta and phi are free to vary (therefore the name), while c(alpha, delta) are constrained to be either c(0, 1) (monotonically increasing curve) or c(1, -1) (monotonically decreasing curve).

This set of functions use a different parameterization from link[drda]{loglogistic2_gradient}. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with eta2 = log(eta) and phi2 = log(phi).

Note that argument theta is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

loglogistic4_fn — *4-parameter log-logistic function*

Description

Evaluate at a particular set of parameters the 4-parameter log-logistic function.

Usage

```
loglogistic4_fn(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The 4-parameter log-logistic function f(x; theta) is defined here as
g(x; theta) = x^eta / (x^eta + phi^eta) f(x; theta) = alpha + delta g(x; theta)

where x >= 0, theta = c(alpha, delta, eta, phi), eta > 0, and phi > 0.

Value

Numeric vector of the same length of x with the values of the log-logistic function.

loglogistic4_gradient — *4-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 4-parameter log-logistic function.

Usage

```
loglogistic4_gradient(x, theta)

loglogistic4_hessian(x, theta)

loglogistic4_gradient_hessian(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The 4-parameter log-logistic function f(x; theta) is defined here as
g(x; theta) = x^eta / (x^eta + phi^eta) f(x; theta) = alpha + delta g(x; theta)

where x >= 0, theta = c(alpha, delta, eta, phi), eta > 0, and phi > 0.

Value

Gradient or Hessian evaluated at the specified point.

---

loglogistic4_gradient_2 *4-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 4-parameter log-logistic function.

Usage

```
loglogistic4_gradient_2(x, theta)

loglogistic4_hessian_2(x, theta)

loglogistic4_gradient_hessian_2(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha, delta, eta, phi).

Details

The 4-parameter log-logistic function f(x; theta) is defined here as
g(x; theta) = x^eta / (x^eta + phi^eta) f(x; theta) = alpha + delta g(x; theta)

where x >= 0, theta = c(alpha, delta, eta, phi, nu), eta > 0, and phi > 0.

This set of functions use a different parameterization from link[drda]{loglogistic4_gradient}. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with eta2 = log(eta) and phi2 = log(phi).

Note that argument theta is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

---

loglogistic5_fn — *5-parameter log-logistic function*

Description

Evaluate at a particular set of parameters the 5-parameter log-logistic function.

Usage

```
loglogistic5_fn(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the five parameters in the form c(alpha, delta, eta, phi, nu).

Details

The 5-parameter log-logistic function f(x; theta) is defined here as
g(x; theta) = (x^eta / (x^eta + nu * phi^eta))^(1 / nu) f(x; theta) = alpha + delta g(x; theta)

where x >= 0, theta = c(alpha, delta, eta, phi, nu), eta > 0, phi > 0, and nu > 0.

Parameter alpha is the value of the function when x = 0. Parameter delta is the (signed) height of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi is related to the mid-value of the function. Parameter nu affects near which asymptote maximum growth occurs.

Value

Numeric vector of the same length of x with the values of the log-logistic function.

loglogistic5_gradient *5-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 5-parameter log-logistic function.

Usage

```
loglogistic5_gradient(x, theta)

loglogistic5_hessian(x, theta)

loglogistic5_gradient_hessian(x, theta)
```

Arguments

x	numeric vector at which the function is to be evaluated.
theta	numeric vector with the five parameters in the form c(alpha, delta, eta, phi, nu).

Details

The 5-parameter log-logistic function `f(x; theta)` is defined here as
`g(x; theta) = (x^eta / (x^eta + nu * phi^eta))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `x >= 0`, `theta = c(alpha, delta, eta, phi, nu)`, `eta > 0`, `phi > 0`, and `nu > 0`.

Value

Gradient or Hessian evaluated at the specified point.

loglogistic5_gradient_2 *5-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 5-parameter log-logistic function.

Usage

```
loglogistic5_gradient_2(x, theta)

loglogistic5_hessian_2(x, theta)

loglogistic5_gradient_hessian_2(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the five parameters in the form `c(alpha, delta, eta, phi, nu)`.

Details

The 5-parameter log-logistic function `f(x; theta)` is defined here as
`g(x; theta) = (x^eta / (x^eta + nu * phi^eta))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `x >= 0`, `theta = c(alpha, delta, eta, phi, nu)`, `eta > 0`, `phi > 0`, and `nu > 0`.

This set of functions use a different parameterization from `link[drda]{loglogistic5_gradient}`. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with `eta2 = log(eta)`, `phi2 = log(phi)`, and `nu2 = log(nu)`.

Note that argument `theta` is on the original scale and not on the log scale.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

`loglogistic6_fn` *6-parameter log-logistic function*

Description

Evaluate at a particular set of parameters the 6-parameter log-logistic function.

Usage

```
loglogistic6_fn(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the six parameters in the form `c(alpha, delta, eta, phi, nu, xi)`.

Details

The 6-parameter log-logistic function `f(x; theta)` is defined here as
`g(x; theta) = (x^eta / (xi * x^eta + nu * phi^eta))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `x >= 0`, `theta = c(alpha, delta, eta, phi, nu, xi)`, `eta > 0`, `phi > 0`, `nu > 0`, and `xi > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

Parameter `alpha` is the value of the function when `x = 0`. Parameter `delta` affects the value of the function when `x -> Inf`. Parameter `eta` represents the steepness (growth rate) of the curve. Parameter `phi` is related to the mid-value of the function. Parameter `nu` affects near which asymptote maximum growth occurs. Parameter `xi` affects the value of the function when `x -> Inf`.

Note: The 6-parameter log-logistic function is over-parameterized and non-identifiable from data. It is available only for theoretical research.

Value

Numeric vector of the same length of `x` with the values of the log-logistic function.

`loglogistic6_gradient` *6-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 6-parameter log-logistic function.

Usage

```
loglogistic6_gradient(x, theta)

loglogistic6_hessian(x, theta)

loglogistic6_gradient_hessian(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the six parameters in the form `c(alpha, delta, eta, phi, nu, xi)`.

Details

The 6-parameter log-logistic function `f(x; theta)` is defined here as
`g(x; theta) = (x^eta / (xi * x^eta + nu * phi^eta))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `x >= 0`, `theta = c(alpha, delta, eta, phi, nu, xi)`, `eta > 0`, `phi > 0`, `nu > 0`, and `xi > 0`. When `delta` is positive (negative) the curve is monotonically increasing (decreasing).

Note: The 6-parameter log-logistic function is over-parameterized and non-identifiable from data. It is available only for theoretical research.

Value

Gradient or Hessian evaluated at the specified point.

`loglogistic6_gradient_2` *6-parameter log-logistic function gradient and Hessian*

Description

Evaluate at a particular set of parameters the gradient and Hessian of the 6-parameter log-logistic function.

Usage

```
loglogistic6_gradient_2(x, theta)

loglogistic6_hessian_2(x, theta)

loglogistic6_gradient_hessian_2(x, theta)
```

Arguments

`x`	numeric vector at which the function is to be evaluated.
`theta`	numeric vector with the six parameters in the form `c(alpha, delta, eta, phi, nu, xi)`.

Details

The 6-parameter log-logistic function `f(x; theta)` is defined here as
`g(x; theta) = (x^eta / (xi * x^eta + nu * phi^eta))^(1 / nu)` `f(x; theta) = alpha + delta g(x; theta)`

where `x` >= `0`, `theta` = `c(alpha, delta, eta, phi, nu, xi)`, `eta` > `0`, `phi` > `0`, `nu` > `0`, and `xi` > `0`.

This set of functions use a different parameterization from `link[drda]{loglogistic6_gradient}`. To avoid the non-negative constraints of parameters, the gradient and Hessian computed here are for the function with `eta2` = `log(eta)`, `phi2` = `log(phi)`, `nu2` = `log(nu)`, and `xi2` = `log(xi)`.

Note that argument `theta` is on the original scale and not on the log scale.

Note: The 6-parameter log-logistic function is over-parameterized and non-identifiable from data. It is available only for theoretical research.

Value

Gradient or Hessian of the alternative parameterization evaluated at the specified point.

naac *Area above the curve*

Description

Evaluate the normalized area above the curve (NAAC).

Usage

```
naac(object, xlim, ylim)
```

Arguments

`object`	fit object as returned by `drda`.
`xlim`	numeric vector of length 2 with the lower and upped bound of the integration interval. Default is `c(-10, 10)` for the logistic function or `c(0, 1000)` for the log-logistic function.
`ylim`	numeric vector of length 2 with the lower and upped bound of the allowed function values. Default is `c(0, 1)`.

Details

The area under the curve (AUC) is the integral of the chosen model `y(x; theta)` with respect to `x`.

In real applications the response variable is usually contained within a known interval. For example, if our response represents relative viability against a control compound, the curve is expected to be between 0 and 1. Let `ylim = c(yl, yu)` represent the admissible range of our function `y(x; theta)`, that is `yl` is its lower bound and `yu` its upper bound. Let `xlim = c(xl, xu)` represent the admissible range of the predictor variable `x`. For example, when `x` represent the dose, the boundaries are the minimum and maximum doses we can administer.

To make the AUC value comparable between different compounds and/or studies, this function sets a hard constraint on both the `x` variable and the function `y`. The intervals can always be changed if needed.

The integral calculated by this function is of the piece-wise function `f(x; theta)` defined as
`f(x; theta) = yl`, if `y(x; theta) < yl`
`f(x; theta) = y(x; theta)`, if `yl <= y(x; theta) <= yu`
`f(x; theta) = yu`, if `y(x; theta) > yu`

The AUC is finally normalized by its maximum possible value, that is the area of the rectangle with width `xu - xl` and height `yu - yl`.

The normalized area above the curve (NAAC) is simply `NAAC = 1 - NAUC`.

Value

Numeric value representing the normalized area above the curve.

See Also

nauc for the Normalized Area Under the Curve (NAUC).

Examples

```
drda_fit <- drda(response ~ log_dose, data = voropm2)
naac(drda_fit)
naac(drda_fit, xlim = c(6, 8), ylim = c(0.2, 0.5))
```

---

nauc *Area under the curve*

Description

Evaluate the normalized area under the curve (NAUC).

Usage

```
nauc(object, xlim, ylim)
```

Arguments

object	fit object as returned by drda.
xlim	numeric vector of length 2 with the lower and upped bound of the integration interval. Default is `c(-10, 10)` for the logistic function or `c(0, 1000)` for the log-logistic function.
ylim	numeric vector of length 2 with the lower and upped bound of the allowed function values. Default is `c(0, 1)`.

Details

The area under the curve (AUC) is the integral of the chosen model `y(x; theta)` with respect to `x`.

In real applications the response variable is usually contained within a known interval. For example, if our response represents relative viability against a control compound, the curve is expected to be between 0 and 1. Let `ylim = c(yl, yu)` represent the admissible range of our function `y(x; theta)`, that is `yl` is its lower bound and `yu` its upper bound. Let `xlim = c(xl, xu)` represent the admissible range of the predictor variable `x`. For example, when `x` represent the dose, the boundaries are the minimum and maximum doses we can administer.

To make the AUC value comparable between different compounds and/or studies, this function sets a hard constraint on both the `x` variable and the function `y`. The intervals can always be changed if needed.

The integral calculated by this function is of the piece-wise function `f(x; theta)` defined as
`f(x; theta) = yl`, if `y(x; theta) < yl`
`f(x; theta) = y(x; theta)`, if `yl <= y(x; theta) <= yu`
`f(x; theta) = yu`, if `y(x; theta) > yu`

The AUC is finally normalized by its maximum possible value, that is the area of the rectangle with width `xu - xl` and height `yu - yl`.

Value

Numeric value representing the normalized area under the curve.

See Also

naac for the Normalized Area Above the Curve (NAAC).

Examples

```
drda_fit <- drda(response ~ log_dose, data = voropm2)
nauc(drda_fit)
nauc(drda_fit, xlim = c(6, 8), ylim = c(0.2, 0.5))
```

---

plot.drda *Model fit plotting*

Description

Plot maximum likelihood curves fitted with drda.

Usage

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

Arguments

x	drda object as returned by the link[drda]{drda} function.
...	other drda objects or parameters to be passed to the plotting functions. See 'Details'.

Details

This function provides a scatter plot of the observed data, overlaid with the maximum likelihood curve fit. If multiple fit objects from the same family of models are given, they will all be placed in the same plot.

Accepted plotting arguments are:

base character string with the base used for printing the values on the x axis. Accepted values are 10 for base 10, 2 for base 2, e for base e, or n (default) for no log-scale printing.

col curve color(s). By default, up to 9 color-blind friendly colors are provided.

xlab, ylab axis labels.

xlim, ylim the range of x and y values with sensible defaults.

level level of confidence intervals (default is 0.95). Set to zero or a negative value to disable confidence intervals.

midpoint if TRUE (default) shows guidelines associated with the curve mid-point.

plot_data if TRUE (default) shows in the plot the data points used for fitting.

legend_show if TRUE (default) shows the legend.

legend_location character string with custom legend position. See link[graphics]{legend} for possible keywords.

legend custom labels for the legend model names.

show If TRUE (default) a figure is plotted, otherwise the function returns a list with values to create the figure manually.

Value

If argument show is TRUE, no return value. If argument show is FALSE, a list with all plotting data.

---

voropm2 *Vorinostat in OPM-2 cell-line dataset*

Description

A dataset containing dose-response data of drug Vorinostat tested ex-vivo on the OPM-2 cell-line.

Usage

```
voropm2
```

Format

A data frame with 45 rows and 4 variables:

response viability measures normalized using positive and negative controls

dose drug concentrations (nM) used for testing

log_dose natural logarithm of variable dose

weight random weights included only for package demonstration

Index