

Package ‘dr4pl’

September 14, 2018

Type Package

Date 2018-09-12

Title Dose Response Data Analysis using the 4 Parameter Logistic (4pl) Model

Version 1.1.7.1

Depends R (>= 3.1.0)

Description Models the relationship between dose levels and responses in a pharmacological experiment using the 4 Parameter Logistic model. Traditional packages on dose-response modelling such as 'drc' and 'nplr' often draw errors due to convergence failure especially when data have outliers or non-logistic shapes. This package provides robust estimation methods that are less affected by outliers and other initialization methods that work well for data lacking logistic shapes. We provide the bounds on the parameters of the 4PL model that prevent parameter estimates from diverging or converging to zero and base their justification in a statistical principle. These methods are used as remedies to convergence failure problems. Gadagkar, S. R. and Call, G. B. (2015) <doi:10.1016/j.vascn.2014.08.006> Ritz, C. and Baty, F. and Streibig, J. C. and hard, D. (2015) <doi:10.1371/journal.pone.0146021>.

License GPL (>= 2)

LazyData TRUE

URL <https://bitbucket.org/dittmerlab/dr4pl>

BugReports <https://bitbucket.org/dittmerlab/dr4pl/issues?status=new&status=open>

RdMacros Rdpack

RoxygenNote 6.0.1

Imports ggplot2, Matrix, matrixcalc, tensor, Rdpack

Suggests drc, devtools, roxygen2, testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Justin T. Landis [aut, cre],
Hyowon An [aut],
Aubrey G. Bailey [aut]

Maintainer Justin T. Landis <jtlandis314@gmail.com>

Repository CRAN

Date/Publication 2018-09-13 23:00:03 UTC

R topics documented:

coef.dr4pl	3
confint.dr4pl	3
dr4pl	4
dr4plEst	7
drc_error_1	8
drc_error_2	9
drc_error_3	9
drc_error_4	9
FindHillBounds	10
FindInitialParms	11
FindLogisticGrids	11
gof.dr4pl	12
IC	13
MeanResponse	14
OutlierDetection	14
plot.dr4pl	15
print.dr4pl	17
print.summary.dr4pl	17
sample_data_1	18
sample_data_10	18
sample_data_11	19
sample_data_12	19
sample_data_13	19
sample_data_2	20
sample_data_3	20
sample_data_4	20
sample_data_5	21
sample_data_6	21
sample_data_7	21
sample_data_8	22
sample_data_9	22
summary.dr4pl	22
vcov.dr4pl	23

Index

25

coef.dr4pl	<i>Obtain coefficients of a 4PL model</i>
------------	---

Description

This function obtains the coefficients of a 4PL model. Estimates of the four parameters, the upper asymptote, IC50, slope and lower asymptote, are returned.

Usage

```
## S3 method for class 'dr4pl'  
coef(object, ...)
```

Arguments

object	A 'dr4pl' object
...	arguments passed to coef

Value

A vector of parameters

Examples

```
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_2) # Fit a 4PL model to data  
coef(obj.dr4pl) # Print parameter estimates  
  
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_3) # Fit a 4PL model to data  
coef(obj.dr4pl) # Print parameter estimates
```

confint.dr4pl	<i>Fit a 4 parameter logistic (4PL) model to dose-response data.</i>
---------------	--

Description

Compute the approximate confidence intervals of the parameters of a 4PL model based on the asymptotic normality of least squares estimators.

Usage

```
## S3 method for class 'dr4pl'  
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	An object of the dr4pl class
parm	Parameters of a 4PL model
level	Confidence level
...	Other parameters to be passed

Details

This function computes the approximate confidence intervals of the true parameters of a 4PL model based on the asymptotic normality of the least squares estimators in nonlinear regression. The Hessian matrix is used to obtain the second order approximation to the sum-of-squares loss function. Please refer to Subsection 5.2.2 of Seber and Wild (1989).

Value

A matrix of the confidence intervals in which each row represents a parameter and each column represents the lower and upper bounds of the confidence intervals of the corresponding parameters.

References

Seber GAF, Wild CJ (1989). *Nonlinear regression*, series Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York. ISBN 0-471-61760-1, doi: [10.1002/0471725315](https://doi.org/10.1002/0471725315), <http://dx.doi.org.libproxy.lib.unc.edu/10.1002/0471725315>.

Examples

```
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_1) # Fit a 4PL model to data

## Use the data 'sample_data_1' to obtain confidence intervals.
confint(obj.dr4pl) # 95% confidence intervals
confint(obj.dr4pl, level = 0.99) # 99% confidence intervals

theta <- FindInitialParms(x = sample_data_1$Dose, y = sample_data_1$Response)

# Use the same data 'sample_data_1' but different parameter estimates to obtain
# confidence intervals.
confint(obj.dr4pl, parm = theta)
```

Description

This function fits a 4PL model to dose-response data. Users can obtain fitted parameter estimates as return values. Using auxiliary functions provided by this R package, users can plot a fitted dose-response curve and obtain confidence intervals of true parameters. In addition, the goodness-of-fit test for model adequacy of the 4PL models can be performed when replicates are available for each dose level.

Usage

```
dr4pl(...)

## S3 method for class 'formula'
dr4pl(formula, data = list(), init.parm = NULL,
      trend = "auto", method.init = "Mead", method.robust = NULL,
      method.optim = "Nelder-Mead", use.Hessian = FALSE, level = 0.9999,
      failure.message = FALSE, ...)

## S3 method for class 'data.frame'
dr4pl(data, dose, response, init.parm = NULL,
      trend = "auto", method.init = "Mead", method.robust = NULL,
      method.optim = "Nelder-Mead", use.Hessian = FALSE, level = 0.9999,
      failure.message = FALSE, ...)

## Default S3 method:
dr4pl(dose, response, init.parm = NULL, trend = "auto",
      method.init = "Mead", method.robust = NULL,
      method.optim = "Nelder-Mead", use.Hessian = FALSE, level = 0.9999,
      failure.message = FALSE, ...)
```

Arguments

...	Further arguments to be passed to <code>constrOptim</code> .
<code>formula</code>	Symbolic description of the model to be fit. Either of the form <code>'response ~ dose'</code> or as a data frame with response values in first column and dose values in second column.
<code>data</code>	Data frame containing variables in the model.
<code>init.parm</code>	Vector of initial parameters to be optimized in the model.
<code>trend</code>	Indicator of whether a dose-response curve is a decreasing $\theta[3] < 0$ or increasing curve $\theta[3] > 0$. The default is "auto" which indicates that the trend of the curve is automatically determined by data. The option "decreasing" will impose a restriction $\theta[3] \leq 0$ while the option "increasing" will impose a restriction $\theta[3] \geq 0$ in an optimization process.
<code>method.init</code>	Method of obtaining initial values of the parameters. If this parameter is left unassigned, a default "Mead" method will be used. Assign "logistic" to use the logistic method.

<code>method.robust</code>	Parameter to select loss function for the robust estimation method to be used to fit a model. The argument <code>NULL</code> indicates the sum of squares loss, "absolute" indicates the absolute deviation loss, "Huber" indicates Huber's loss and "Tukey" indicates Tukey's biweight loss.
<code>method.optim</code>	Method of optimization of the loss function specified by <code>method.robust</code> . This function argument is directly passed to the function <code>constrOptim</code> which is provided in the <code>base</code> package of R.
<code>use.Hessian</code>	Indicator of whether the Hessian matrix (TRUE) or the gradient vector is used in the Hill bounds.
<code>level</code>	Confidence level to be used in Hill bounds computation.
<code>failure.message</code>	Indicator of whether a message indicating attainment of the Hill bounds and possible resolutions will be printed to the console (TRUE) or hidden (FALSE).
<code>dose</code>	Vector of dose levels
<code>response</code>	Vector of responses

Details

This function fits a 4 parameter logistic (4PL) model to dose-response data. A formula of the model is

$$\theta[1] + (\theta[4] - \theta[1]) / (1 + (z/\theta[2])^{\theta[3]})$$

`method.init` specifies an initialization method to get initial parameter estimates based on data. The currently supported initialization methods are "logistic" and "Mead". For further details, see the vignette.

`method.optim` specifies an optimization method to be used in "constrOptim" function. The currently supported optimization techniques include "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN" and "Brent". For further details, see the help page of `optim`.

`method.robust` chooses a robust estimation method among 4 methods. The method of estimation is usually identified by the loss function of the method. This package supports 4 types of loss functions: sum-of-squares loss, absolute deviation loss, Huber's loss and Tukey's biweight loss. Each of loss function is explained in detail in the vignette.

Value

A 'dr4pl' object for which "confint", "gof", "print" and "summary" methods are implemented. For details, see the help page of each method. For example, type `?confint.dr4pl` to obtain the confidence intervals of parameters of the 'dr4pl' object.

Methods (by class)

- `formula`: General 4PL model fitting function for analysis of dose-response relation.
- `data.frame`: Method for when formula argument is missing. dose and response arguments are necessary
- `default`: Used in the default case, supplying a single dose and response variable

Author(s)

Hyowon An, <ahwbest@gmail.com>
 Justin T. Landis, <jtlandis314@gmail.com>
 Aubrey G. Bailey, <aubreybailey@gmail.com>

See Also

[confint.dr4pl](#), [gof.dr4pl](#), [print.dr4pl](#), [summary.dr4pl](#)

Examples

```
##Assign method.init = "logistic" to use logistic method of estimation.
##default method
a <- dr4pl(dose = sample_data_1$Dose,
           response = sample_data_1$Response,
           method.init = "logistic")
plot(a)

##Use default or Assign method.init = "Mead" to use Mead's method of estimation.
# Use method.robust to select desired loss function
# formula method
b <- dr4pl(formula = Response~Dose,
           data = sample_data_4,
           method.init = "Mead",
           method.robust = "Tukey" )
plot(b)

#data.frame method
c <- dr4pl(data = sample_data_10,
           dose = Dose,
           response = Response)
plot(c)

##compatible with ggplot
library(ggplot2) #load ggplot2
c <- dr4pl(Response~Dose,
           data = drc_error_2,
           method.optim = "CG",
           trend = "decreasing" )
d <- plot(c, x.breaks = c(.00135, .0135, .135, 1.35, 13.5))
d + theme_grey()
```

dr4plEst

Private function to fit the 4PL model to dose-response data

Description

Private function that actually fits the 4PL model to data. If the Hill bounds are attained at the end of optimization processes, then an indicator of convergence failure so that [dr4pl.default](#) can look for a remedy for convergence failure.

Usage

```
dr4plEst(dose, response, init.parm, trend, method.init, method.optim,
         method.robust, use.Hessian, level)
```

Arguments

dose	Vector of dose levels
response	Vector of responses
init.parm	Vector of initial parameters of the 4PL model supplied by a user.
trend	Indicator of whether a dose-response curve is a decreasing $\theta[3] < 0$ or increasing curve $\theta[3] > 0$. The default is "auto" which indicates that the trend of the curve is automatically determined by data. The option "decreasing" will impose a restriction $\theta[3] \leq 0$ while the option "increasing" will impose a restriction $\theta[3] \geq 0$ in an optimization process.
method.init	Method of obtaining initial values of the parameters. Should be one of "logistic" for the logistic method or "Mead" for the Mead method. The default option is the Mead method.
method.optim	Method of optimization of the parameters. This argument is directly delivered to the <code>constrOptim</code> function provided in the "base" package of R.
method.robust	Parameter to select loss function for the robust estimation method to be used to fit a model. The argument NULL indicates the sum of squares loss, "absolute" indicates the absolute deviation loss, "Huber" indicates Huber's loss and "Tukey" indicates Tukey's biweight loss.
use.Hessian	Indicator of whether the Hessian matrix (TRUE) or the gradient vector is used in the Hill bounds.
level	Confidence level to be used in Hill bounds computation.

Value

List of final parameter estimates, name of robust estimation, loss value and so on.

drc_error_1

Single High Outlier

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These all have numerical errors in other dose-response curve-packages, but not using these methods. This data set exemplifies the case of a single extreme outlier of one dose measurement.

Examples

```
a <- dr4pl(Response~Dose, data = drc_error_1, method.init = "logistic", method.robust = "Tukey")
plot(a)
```

drc_error_2

Multiple High Outliers at Different measurements

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These all have numerical errors in other dose-response curve-packages, but not using these methods. This data set exemplifies the case of multiple outliers as well as a small number of observations per dose measurement.

Examples

```
a <- dr4pl(Response~Dose, data = drc_error_2, trend = "decreasing", method.optim = "CG")
plot(a)
```

drc_error_3

Support Problem and Outliers at a Single Dose Level

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These all have numerical errors in other dose-response curve-packages, but not using these methods. This data set exemplifies the case of multiple outliers at a single dose measurement as well as the support problem.

Examples

```
a <- dr4pl(Response~Dose, data = drc_error_3, method.init = "Mead", method.robust = "Huber")
plot(a)
```

drc_error_4

Support Problem

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These all have numerical errors in other dose-response curve-packages, but not using these methods. This data set exemplifies the support problem.

Examples

```
a <- dr4pl(Response~Dose, data = drc_error_4, method.init = "logistic")
plot(a)
```

FindHillBounds

FindHillBounds

Description

Compute the Hill bounds based on initial parameter estimates and data.

Usage

```
FindHillBounds(x, y, theta, use.Hessian = FALSE, level = 0.9999)
```

Arguments

x	Vector of doses.
y	Vector of responses.
theta	Parameters of a 4PL model.
use.Hessian	Indicator of whether the Hessian matrix (TRUE) or the gradient vector is used in confidence interval computation.
level	Confidence level to be used in computing the Hill bounds.

Details

This function computes the Hill bounds based on initial parameter estimates and data. It basically computes the confidence intervals of the true parameters based on the variance-covariance matrix of a given initial parameter estimates. The half of a hessian matrix is used as a variance-covariance matrix. If matrix inversion of the variance-covariance matrix is infeasible, a variation of the method in Wang et al. (2010) is used. The parameter level is only for simulation.

Value

Data frame whose first column represents the bounds on the IC50 parameter in log 10 scale and second column represents the bounds on the slope parameter.

Author(s)

Hyowon An, <ahwbest@gmail.com>.

References

Higham NJ (2002). “Computing the nearest correlation matrix—a problem from finance.” *IMA J. Numer. Anal.*, **22**(3), 329–343. ISSN 0272-4979, doi: [10.1093/imanum/22.3.329](https://doi.org/10.1093/imanum/22.3.329), <http://dx.doi.org.libproxy.lib.unc.edu/10.1093/imanum/22.3.329>. Wang Y, Jadhav A, Southal N, Huang R, Nguyen DT (2010). “A grid algorithm for high throughput fitting of dose-response curve data.” *Curr Chem Genomics*, **4**, 57–66.

See Also

[FindInitialParms](#), [FindLogisticGrids](#).

FindInitialParms *FindInitialParms*

Description

Find initial parameter estimates for a 4PL model.

Usage

```
FindInitialParms(x, y, trend = "auto", method.init = "Mead",
  method.robust = NULL)
```

Arguments

x	Vector of dose levels
y	Vector of responses
trend	Indicator of whether the curve is a decreasing $\theta[3] < 0$ or increasing curve $\theta[3] > 0$. See dr4pl for detailed explanation.
method.init	Method of obtaining initial values of the parameters. See dr4pl for detailed explanation.
method.robust	Parameter to select loss function for the robust estimation method to be used to fit a model. See dr4pl for detailed explanation.

Value

Initial parameter estimates of a 4PL model in the order of the upper asymptote, IC50, Slope and lower asymptote parameters.

FindLogisticGrids *FindLogisticGrids*

Description

Compute the grids on the upper and lower asymptote parameters for the logistic method based on initial parameter estimates and data.

Usage

```
FindLogisticGrids(x, y, retheta.init, use.Hessian = FALSE)
```

Arguments

x	Vector of doses.
y	Vector of responses.
retheta.init	Parameters of a 4PL model among which the EC50 parameter is in the log 10 dose scale.
use.Hessian	Indicator of whether the Hessian matrix (TRUE) or the gradient vector is used in confidence interval computation.

Details

This function computes the grids on the upper and lower asymptote parameters based on initial parameter estimates and data. It basically computes the confidence intervals of the true parameters based on the variance-covariance matrix of the given initial parameter estimates. If matrix inversion of the variance-covariance matrix is infeasible, a variation of the method in Wang et al. (2010) is used.

Value

Data frame whose first column represents the grid on the upper asymptote parameter and second column represents the grid on the lower asymptote.

Author(s)

Hyowon An

References

Wang Y, Jadhav A, Southal N, Huang R, Nguyen DT (2010). "A grid algorithm for high throughput fitting of dose-response curve data." *Curr Chem Genomics*, **4**, 57–66.

See Also

FindHillBounds, FindInitialParms

gof.dr4pl

Perform the goodness-of-fit (gof) test for the 4PL model.

Description

Perform the goodness-of-fit (gof) test for the 4PL model when there are at least two replicates for each dose level.

Usage

```
gof.dr4pl(object, n.signif.digit = 4)
```

Arguments

`object` An object of the `dr4pl` class.

`n.signif.digit` Number of significant digits after the decimal point to be printed. The default value is 4, but users can change the value on their own.

Details

Perform a goodness-of-fit (gof) test for the goodness of the 4PL models for dose-response data. There should be at least two replicates at each dose level. The test statistic follows the F distribution with degrees of freedom $(n - 4)$ and $(N - n)$ where N stands for the total number of observations and n stands for the number of dose levels. For detailed explanation of the method, please refer to Subsection 2.1.5 of Seber and Wild (1989).

Value

A list of results in the order of a F-statistic value, p-value and a degree of freedom.

References

Seber GAF, Wild CJ (1989). *Nonlinear regression*, series Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York. ISBN 0-471-61760-1, doi: [10.1002/0471725315](https://doi.org/10.1002/0471725315), <http://dx.doi.org.libproxy.lib.unc.edu/10.1002/0471725315>.

Examples

```
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_4) # Fit a 4PL model to data
gof.dr4pl(obj.dr4pl) # Print the goodness-of-fit test results
```

IC *Obtain Inhibitory Concentrations (IC) of a dose-response curve*

Description

This function obtains estimates of the IC's of a dose-response curve. Typically the IC50 parameter is of interest, but sometimes IC10 or IC90 are important aspects of a dose-response curve. By controlling the function argument, a user can obtain the IC's at various levels.

Usage

```
IC(object, inhib.percent)
```

Arguments

`object` Object of the class 'dr4pl' for which the IC values are obtained

`inhib.percent` Inhibited percentages at which the IC values are obtained

Value

IC values at the inhibited percentages provided by the argument `inhib.percent`

Examples

```
data.test <- data.frame(x = c(0.0001, 0.001, 0.01, 0.1, 1),
                       y = c(10, 9, 5, 1, 0))
obj.dr4pl <- dr4pl(y ~ x,
                  data = data.test)
IC(obj.dr4pl, inhib.percent = c(10, 90))

obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_4) # Fit a 4PL model to data
IC(obj.dr4pl, inhib.percent = c(10, 50, 90))
```

MeanResponse	<i>Compute an estimated mean response.</i>
--------------	--

Description

Compute an estimated mean response.

Usage

```
MeanResponse(x, theta)
```

Arguments

<code>x</code>	Dose levels.
<code>theta</code>	Parameters of the 4PL model.

Value

Predicted response values.

OutlierDetection	<i>Detect outliers by the method of Motulsky and Brown (2006).</i>
------------------	--

Description

Detect outliers by the method of Motulsky and Brown (2006).

Usage

```
OutlierDetection(residuals)
```

Arguments

residuals Vector of residuals from a robust fit.

Details

This function detects outliers from a vector of residuals obtained from a robust fit. The method used here is the same with Motulsky and Brown (2006) except that the median absolute deviation is used instead of the sample quantile based estimator suggested in that paper. Based on the False Discovery Rate (FDR) a set of multiple outliers that have lower FDR's than a threshold are reported.

Value

Vector of indices of outliers in the input vector of residuals

Author(s)

Hyowon An

References

Motulsky HJ, Brown RE (2006). "Detecting outliers when fitting data with nonlinear regression - a new method based on robust nonlinear regression and the false discovery rate." *BMC Bioinformatics*, **7**, 123.

plot.dr4pl

Make a plot of a 4PL model curve and data

Description

This function displays a dose-response curve and data. As a default, the x-axis represents dose levels in log 10 scale and the y-axis represents responses. The black solid line represents a dose-response curve. The blue filled circles represent data points and red triangles represent outliers.

Usage

```
## S3 method for class 'dr4pl'
plot(x, type.curve = "all",
     text.title = "Dose-response plot", text.x = "Dose", text.y = "Response",
     indices.outlier = NULL, breaks.x = NULL, breaks.y = NULL, ...)
```

Arguments

x 'dr4pl' object whose data and mean response function will be plotted.

type.curve Indicator of the type of a dose-response curve. "all" indicates that data and a curve will be plotted while "data" indicates that only data will be plotted.

text.title Character string for the title of a plot with a default set to "Dose response plot".

text.x Character string for the x-axis of the plot with a default set to "Dose".

text.y Character string for the y-axis of the plot with a default set to "Response".

indices.outlier Pass a vector indicating all indices which are outliers in the data.

breaks.x Vector of desired break points for the x-axis

breaks.y Vector of desired break points for the y-axis

... All arguments that can normally be passed to ggplot.

Author(s)

Hyowon An, <ahwbest@gmail.com>
 Justin T. Landis, <jtlandis314@gmail.com>
 Aubrey G. Bailey, <aubreybailey@gmail.com>

Examples

```
dr4pl.1 <- dr4pl(Response ~ Dose, data = sample_data_1)

plot(dr4pl.1)

## Able to further edit plots.
library(ggplot2) #needed to change color to green
dr4pl.1 <- dr4pl(Response ~ Dose,
                 data = sample_data_1,
                 text.title = "Sample Data Plot")

a <- plot(dr4pl.1)
a + geom_point(color = "green", size = 5)

## Bring attention to outliers using parameter indices.outlier.
dr4pl.3 <- dr4pl(Response ~ Dose,
                 data = drc_error_3,
                 method.init = "Mead",
                 method.robust = "absolute")
plot(dr4pl.3, indices.outlier = c(90, 101))

## Change the plot title default with parameter text.title.
dr4pl.1 <- dr4pl::dr4pl(Response ~ Dose,
                       data = sample_data_1)
plot(dr4pl.1, text.title = "My New Dose Response plot")

##Change the labels of the x and y axis to your need
library(drc) # Needed to load 'decontaminants' data set
data.hpc <- subset(decontaminants, group %in% "hpc")
dr4pl.hpc <- dr4pl(count~conc, data = data.hpc)
plot(dr4pl.hpc,
     text.title = "hpc Decontaminants Plot",
     text.x = "Concentration",
     text.y = "Count")
```

print.dr4pl	<i>Print the dr4pl object to screen.</i>
-------------	--

Description

Print the dr4pl object to screen.

Usage

```
## S3 method for class 'dr4pl'  
print(object, ...)
```

Arguments

object	a dr4pl object to be printed
...	all normally printable arguments

Examples

```
ryegrass.dr4pl <- dr4pl(Response ~ Dose,  
                        data = sample_data_1)  
print(ryegrass.dr4pl)  
  
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_5)  
print(obj.dr4pl)
```

print.summary.dr4pl	<i>Print the dr4pl object summary to screen.</i>
---------------------	--

Description

Print the dr4pl object summary to screen.

Usage

```
## S3 method for class 'summary.dr4pl'  
print(object, ...)
```

Arguments

object	a dr4pl object to be summarized
...	all normally printable arguments

Examples

```
library(drc) # Needed for the data set 'ryegrass'
dr4pl.ryegrass <- dr4pl(root1 ~ conc, data = ryegrass)
print(summary(dr4pl.ryegrass))

dr4pl.7 <- dr4pl(Response ~ Dose, data = sample_data_7)
print(summary(dr4pl.7))
```

sample_data_1	<i>sample_data_1</i>
---------------	----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_1)
plot(a)
```

sample_data_10	<i>sample_data_10</i>
----------------	-----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_10)
plot(a)
```

sample_data_11	<i>sample_data_11</i>
----------------	-----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_11)
plot(a)
```

sample_data_12	<i>sample_data_12</i>
----------------	-----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_12)
plot(a)
```

sample_data_13	<i>sample_data_13</i>
----------------	-----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_13)
plot(a)
```

`sample_data_2`*sample_data_2*

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_2)
plot(a)
```

`sample_data_3`*sample_data_3*

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_3)
plot(a)
```

`sample_data_4`*sample_data_4*

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_4)
plot(a)
```

sample_data_5	<i>sample_data_5</i>
---------------	----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_5)
plot(a)
```

sample_data_6	<i>sample_data_6</i>
---------------	----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_6)
plot(a)
```

sample_data_7	<i>sample_data_7</i>
---------------	----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_7)
plot(a)
```

sample_data_8	<i>sample_data_8</i>
---------------	----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_8)
plot(a)
```

sample_data_9	<i>sample_data_9</i>
---------------	----------------------

Description

These are a handful of experimentally derived datasets from the wet-laboratory. These may or may not have numerical errors in other dose-response curve-packages, but definitely not using these methods.

Examples

```
a <- dr4pl(Response~Dose, data = sample_data_9)
plot(a)
```

summary.dr4pl	<i>summary</i>
---------------	----------------

Description

Print the summary of a dr4pl object.

Usage

```
## S3 method for class 'dr4pl'
summary(object, ...)
```

Arguments

object	a dr4pl object to be summarized
...	all normal summary arguments

Examples

```
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_5) # Fit a 4PL model to data
summary(obj.dr4pl)
```

```
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_6) # Fit a 4PL model to data
summary(obj.dr4pl)
```

vcov.dr4pl	<i>Obtain the variance-covariance matrix of the parameter estimators of a 4PL model.</i>
------------	--

Description

This function obtains the variance-covariance matrix of the parameter estimators of a 4PL model. The variance-covariance matrix returned by this function can be used to compute the standard errors and confidence intervals for statistical inference.

Usage

```
## S3 method for class 'dr4pl'
vcov(object, ...)
```

Arguments

object	An object of the dr4pl class.
...	Other function arguments to be passed to the default 'vcov' function.

Details

This function obtains the variance-covariance matrix of the parameter estimators of a 4PL model. The Hessian matrix is used to obtain the second order approximation to the sum-of-squares loss function, and then the standard errors are computed as the square roots of the half of the Hessian matrix. Please refer to Subsection 5.2.2 of Seber and Wild (1989).

Value

The variance-covariance matrix of the parameter estimators of a 4PL model whose columns are in the order of the upper asymptote, IC50, slope and lower asymptote from left to right and whose rows are in the same order.

References

Seber GAF, Wild CJ (1989). *Nonlinear regression*, series Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York. ISBN 0-471-61760-1, doi: [10.1002/0471725315](https://doi.org/10.1002/0471725315), <http://dx.doi.org.libproxy.lib.unc.edu/10.1002/0471725315>.

Examples

```
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_1) # Fit a 4PL model to data  
vcov(obj.dr4pl) # Variance-covariance matrix of the parameters
```

```
obj.dr4pl <- dr4pl(Response ~ Dose, data = sample_data_2) # Fit a 4PL model to data  
vcov(obj.dr4pl) # Variance-covariance matrix of the parameters
```


Index

*Topic **sample_data**

- drc_error_1, 8
- drc_error_2, 9
- drc_error_3, 9
- drc_error_4, 9
- sample_data_1, 18
- sample_data_10, 18
- sample_data_11, 19
- sample_data_12, 19
- sample_data_13, 19
- sample_data_2, 20
- sample_data_3, 20
- sample_data_4, 20
- sample_data_5, 21
- sample_data_6, 21
- sample_data_7, 21
- sample_data_8, 22
- sample_data_9, 22

coef.dr4pl, 3

confint.dr4pl, 3, 7

constrOptim, 6

dr4pl, 4, 11

dr4pl-package (dr4pl), 4

dr4pl.data.frame (dr4pl), 4

dr4pl.default, 7

dr4pl.default (dr4pl), 4

dr4pl.formula (dr4pl), 4

dr4plEst, 7

drc_error_1, 8

drc_error_2, 9

drc_error_3, 9

drc_error_4, 9

FindHillBounds, 10

FindInitialParms, 10, 11

FindLogisticGrids, 10, 11

gof.dr4pl, 7, 12

IC, 13

MeanResponse, 14

optim, 6

OutlierDetection, 14

plot.dr4pl, 15

print.dr4pl, 7, 17

print.summary.dr4pl, 17

sample_data_1, 18

sample_data_10, 18

sample_data_11, 19

sample_data_12, 19

sample_data_13, 19

sample_data_2, 20

sample_data_3, 20

sample_data_4, 20

sample_data_5, 21

sample_data_6, 21

sample_data_7, 21

sample_data_8, 22

sample_data_9, 22

summary.dr4pl, 7, 22

vcov.dr4pl, 23