Package 'AgroReg'

July 21, 2025

Type Package

Title Regression Analysis Linear and Nonlinear for Agriculture

Version 1.2.11

Date 2025-06-22

Maintainer Gabriel Danilo Shimizu <gabrield.shimizu@gmail.com>

Description Linear and nonlinear regression analysis common in agricultural science articles (Archontoulis & Miguez (2015). <doi:10.2134/agronj2012.0506>). The package includes polynomial, exponential, gaussian, logistic, logarithmic, segmented, non-parametric models, among others. The functions return the model coefficients and their respective p values, coefficient of determination, root mean square error, AIC, BIC, as well as graphs with the equations automatically.

License GPL (>= 2)

URL https://fisher.uel.br/AgroReg_shiny/,

https://fisher.uel.br/AgroReg_shiny.pt/

Imports drc, ggplot2, boot, minpack.lm, dplyr, rcompanion, broom, egg, purrr

Depends R (>= 3.6)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Gabriel Danilo Shimizu [aut, cre] (ORCID: <https://orcid.org/0000-0001-8524-508X>), Leandro Simoes Azeredo Goncalves [aut, ctb] (ORCID: <https://orcid.org/0000-0001-9700-9375>)

Repository CRAN

Date/Publication 2025-07-01 20:50:02 UTC

Contents

`adjust_scale` *Utils: Adjust y and x scale*

Description

Adjust y and x scale for chart or charts

Usage

```
adjust_scale(
  plots,
  scale.x = "default",
  limits.x = "default",
  scale.y = "default",
  limits.y = "default"
)
```

Arguments

`plots`	Object of analysis or plot_arrange
`scale.x`	x-axis scale (use vector)
`limits.x`	limits in x-axis (use vector)
`scale.y`	y-axis scale (use vector)
`limits.y`	limits in y-axis (use vector)

Value

Returns the scaled graph

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
a=LM(trat,resp)
b=LL(trat,resp,npar = "LL.3")
a=plot_arrange(list(a,b),gray = TRUE)
adjust_scale(a,scale.y = seq(0,100,10),limits.y = c(0,100))
```

`adjust_scale_x` *Utils: Adjust x scale*

Description

Adjust x scale for chart or charts

Usage

```
adjust_scale_x(plots, scale = "default", limits = "default")
```

Arguments

`plots`	Object of analysis or plot_arrange
`scale`	x-axis scale (use vector)
`limits`	limits in x-axis (use vector)

Value

Returns the scaled graph

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
a=LM(trat,resp)
b=LL(trat,resp,npar = "LL.3")
a=plot_arrange(list(a,b),gray = TRUE)
adjust_scale_x(a,scale = seq(10,40,5),limits = c(10,40))
```

adjust_scale_y *Utils: Adjust y scale*

Description

Adjust y scale for chart or charts

Usage

```
adjust_scale_y(plots, scale = "default", limits = "default")
```

Arguments

plots	Object of analysis or plot_arrange
scale	y-axis scale (use vector)
limits	limits in y-axis (use vector)

Value

Returns the scaled graph

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
a=LM(trat,resp)
b=LL(trat,resp,npar = "LL.3")
a=plot_arrange(list(a,b),gray = TRUE)
adjust_scale_y(a,scale = seq(0,100,10),limits = c(0,100))
```

AM *Analysis: Avhad and Marchetti*

Description

This function performs Avhad and Marchetti regression analysis.

Usage

```
AM(
  trat,
  resp,
  initial = list(alpha, k, n),
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.
`initial`	Starting estimates
`sample.curve`	Provide the number of observations to simulate curvature (default is 1000)
`ylab`	Variable response name (Accepts the *expression*() function)
`xlab`	treatments name (Accepts the *expression*() function)
`theme`	ggplot2 theme (*default* is theme_bw())
`legend.position`	legend position (*default* is "top")
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`r2`	coefficient of determination of the mean or all values (*default* is all)
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width

scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The Avhad e Marchetti model is defined by:

$$y = \alpha \times e^{kx^n}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Avhad, M. R., & Marchetti, J. M. (2016). Mathematical modelling of the drying kinetics of Hass avocado seeds. Industrial Crops and Products, 91, 76-87.

Examples

```
library(AgroReg)
data("granada")
attach(granada)
AM(time,100-WL,initial=list(alpha = 610.9129, k=-1.1810, n=0.1289 ))
```

aristolochia *Dataset: Aristolochia*

Description

The data come from an experiment conducted at the Seed Analysis Laboratory of the Agricultural Sciences Center of the State University of Londrina, in which five temperatures (15, 20, 25, 30 and 35C) were evaluated in the germination of *Aristolochia elegans*. The experiment was conducted in a completely randomized design with four replications of 25 seeds each.

Usage

```
data("aristolochia")
```

Format

data.frame containing data set

`trat` Numeric vector with temperature

`resp` Numeric vector with response

Author(s)

Hugo Roldi Guariz

Examples

```
data(aristolochia)
```

asymptotic *Analysis: Asymptotic, exponential or Logarithmic*

Description

This function performs asymptotic regression analysis.

Usage

```
asymptotic(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
```

```
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation

xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The exponential model is defined by:

$$y = \alpha \times e^{-\beta \cdot x} + \theta$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley and Sons (p. 330).

Examples

```
library(AgroReg)
data("granada")
attach(granada)
asymptotic(time,100-WL)
```

---

asymptotic_i *Analysis: Asymptotic without intercept*

Description

This function performs asymptotic regression analysis without intercept.

Usage

```
asymptotic_i(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  fontfamily = "sans",
  comment = NA,
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.
`sample.curve`	Provide the number of observations to simulate curvature (default is 1000)
`ylab`	Variable response name (Accepts the *expression*() function)
`xlab`	treatments name (Accepts the *expression*() function)
`theme`	ggplot2 theme (*default* is theme_bw())
`legend.position`	legend position (*default* is "top")
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`r2`	coefficient of determination of the mean or all values (*default* is all)
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`scale`	Sets x scale (*default* is none, can be "log")
`textsize`	Font size

pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
fontfamily	Font family
comment	Add text after equation
print.on	Print output

Details

The asymptotic model without intercept is defined by:

$$y = \alpha \times e^{-\beta \cdot x}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley and Sons (p. 330).

Siqueira, V. C., Resende, O., & Chaves, T. H. (2013). Mathematical modelling of the drying of jatropha fruit: an empirical comparison. Revista Ciencia Agronomica, 44, 278-285.

Examples

```
library(AgroReg)
data("granada")
attach(granada)
asymptotic_i(time,100-WL)
```

asymptotic_ineg *Analysis: Asymptotic or Exponential Negative without intercept*

Description

This function performs asymptotic regression analysis without intercept.

Usage

```
asymptotic_ineg(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())

`legend.position`	legend position (*default* is "top")
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`r2`	coefficient of determination of the mean or all values (*default* is all)
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`scale`	Sets x scale (*default* is none, can be "log")
`textsize`	Font size
`pointsize`	shape size
`linesize`	line size
`linetype`	line type
`pointshape`	format point (default is 21)
`fillshape`	Fill shape
`colorline`	Color lines
`round`	round equation
`xname.formula`	Name of x in the equation
`yname.formula`	Name of y in the equation
`comment`	Add text after equation
`fontfamily`	Font family
`print.on`	Print Output

Details

The asymptotic negative model without intercept is defined by:

$$y = \alpha \times e^{-\beta \cdot x}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Siqueira, V. C., Resende, O., & Chaves, T. H. (2013). Mathematical modelling of the drying of jatropha fruit: an empirical comparison. Revista Ciencia Agronomica, 44, 278-285.

Examples

```
library(AgroReg)
data("granada")
attach(granada)
asymptotic_ineg(time,100-WL)
```

`asymptotic_neg` *Analysis: Asymptotic or Exponential Negative*

Description

This function performs asymptotic regression analysis.

Usage

```
asymptotic_neg(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.

sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print Output

Details

The asymptotic model is defined by:

$$y = -\alpha \times e^{-\beta \cdot x} + \theta$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
library(AgroReg)
data("granada")
attach(granada)
asymptotic_neg(time,WL)
```

BC *Analysis: Brain-Cousens*

Description

The 'BC.4' and 'BC.5' logistical models provide Brain-Cousens' modified logistical models to describe u-shaped hormesis. This model was extracted from the 'drc' package.

Usage

```
BC(
  trat,
  resp,
  npar = "BC.4",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  error = "SE",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
```

```
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	Number of model parameters (*default* is BC.4)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	Treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	Legend position (*default* is "top")
r2	Coefficient of determination of the mean or all values (*default* is all)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
error	Error bar (It can be SE - *default*, SD or FALSE)
point	Defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	Shape size
linesize	Line size
linetype	line type
pointshape	Format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print Output

Details

The model function for the Brain-Cousens model (Brain and Cousens, 1989) is

$$y = c + \frac{d - c + fx}{1 + \exp(b(\log(x) - \log(e)))}$$

and it is a five-parameter model, obtained by extending the four-parameter log-logistic model (LL.4 to take into account inverse u-shaped hormesis effects. Fixing the lower limit at 0 yields the four-parameter model

$$y = 0 + \frac{d - 0 + fx}{1 + \exp(b(\log(x) - \log(e)))}$$

used by van Ewijk and Hoekstra (1993).

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the drc package (Ritz et al., 2016)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Ritz, C.; Strebig, J.C. and Ritz, M.C. Package 'drc'. Creative Commons: Mountain View, CA, USA, 2016.

See Also

LL, CD,GP

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
BC(trat,resp)
```

beta_reg *Analysis: Beta*

Description

This function performs beta regression analysis.

Usage

```
beta_reg(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	Treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())

`legend.position`	Legend position (*default* is "top")
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`r2`	Coefficient of determination of the mean or all values (*default* is all)
`point`	Defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`scale`	Sets x scale (*default* is none, can be "log")
`textsize`	Font size
`pointsize`	Shape size
`linesize`	Line size
`linetype`	line type
`pointshape`	Format point (default is 21)
`fillshape`	Fill shape
`colorline`	Color lines
`round`	round equation
`xname.formula`	Name of x in the equation
`yname.formula`	Name of y in the equation
`comment`	Add text after equation
`fontfamily`	Font family
`print.on`	Print output

Details

The beta model is defined by:

$$Y = d \times \{(\frac{X - X_b}{X_o - X_b})(\frac{X_c - X}{X_c - X_o})^{\frac{X_c - X_o}{X_o - X_b}}\}^b$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the aomisc package (Andrea Onofri)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Onofri, A., 2020. The broken bridge between biologists and statisticians: a blog and R package. Statforbiology. http://www.statforbiology.com/tags/aomisc/

Examples

```
library(AgroReg)
X <- c(1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50)
Y <- c(0, 0, 0, 7.7, 12.3, 19.7, 22.4, 20.3, 6.6, 0, 0)
beta_reg(X,Y)
```

`biexponential` *Analysis: Biexponential*

Description

This function performs biexponential regression analysis.

Usage

```
biexponential(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.

sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	Treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	Legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	Coefficient of determination of the mean or all values (*default* is all)
point	Defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	Shape size
linesize	Line size
linetype	line type
pointshape	Format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The biexponential model is defined by:

$$y = A1 \times e^{-e^{lrc1 \cdot x}} + A2 \times e^{-e^{lrc2 \cdot x}}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

See Also

asymptotic_neg

Examples

```
library(AgroReg)
data("granada")
attach(granada)
biexponential(time,WL)
```

CD *Analysis: Cedergreen-Ritz-Streibig*

Description

The 'CRS.4' and 'CRS.5' logistical models provide Brain-Cousens modified logistical models to describe u-shaped hormesis. This model was extracted from the 'drc' package.

Usage

```
CD(
  trat,
  resp,
  npar = "CRS.4",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
```

```
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	Number of model parameters
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The four-parameter model is given by the expression:

$$y = 0 + \frac{d - 0 + f\exp(-1/x)}{1 + \exp(b(\log(x) - \log(e)))}$$

while the five-parameter is:

$$y = c + \frac{d - c + f\exp(-1/x)}{1 + \exp(b(\log(x) - \log(e)))}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the drc package (Ritz et al., 2016)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Ritz, C.; Strebig, J.C.; Ritz, M.C. Package 'drc'. Creative Commons: Mountain View, CA, USA, 2016.

See Also

LL, BC, GP

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
CD(trat,resp)
```

coloredit_arrange *Change the colors of a graph from the plot_arrange function*

Description

Change the colors of a graph from the plot_arrange function

Usage

```
coloredit_arrange(graphs, color = NA)
```

Arguments

graphs	object from a plot_arrange function
color	color curve and point

Value

The function changes the colors of a graph coming from the plot_arrange function

Author(s)

Gabriel Danilo Shimizu

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
graph1=LM(trat,resp)
graph2=LL(trat,resp,npar = "LL.3")
graph=plot_arrange(list(graph1,graph2))
coloredit_arrange(graph,color=c("red","blue"))
```

comparative_model *Analysis: Comparative models*

Description

This function allows the construction of a table and/or graph with the statistical parameters to choose the model from the analysis functions.

Usage

```
comparative_model(models, names_model = NA, plot = FALSE, round.label = 2)
```

Arguments

models	List with objects of type analysis
names_model	Names of the models
plot	Plot in the parameters
round.label	Round label plot

Value

Returns a table and/or graph with the statistical parameters for choosing the model.

Author(s)

Gabriel Danilo Shimizu

Examples

```
library(AgroReg)
data(granada)
attach(granada)
a=LM(time,WL)
b=LL(time,WL)
c=BC(time,WL)
d=weibull(time,WL)
comparative_model(models=list(a,b,c,d),names_model=c("LM","LL","BC","Weibull"))

models <- c("LM1", "LM4", "L3", "BC4","weibull3","mitscherlich", "linear.plateau", "VG")
r <- lapply(models, function(x) {
r <- with(granada, regression(time, WL, model = x))
})
comparative_model(r,plot = TRUE)
```

---

correlation *Graph: Plot correlation*

Description

Correlation analysis function (Pearson or Spearman)

Usage

```
correlation(
  x,
  y,
  method = "pearson",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
```

```
  textsize = 12,
  pointsize = 5,
  pointshape = 21,
  linesize = 0.8,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  ic = TRUE,
  title = NA,
  fontfamily = "sans"
)
```

Arguments

x	Numeric vector with independent variable
y	Numeric vector with dependent variable
method	Method correlation (*default* is Pearson)
ylab	Variable response name (Accepts the *expression*() function)
xlab	Treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
textsize	Axis text size
pointsize	Point size
pointshape	shape format
linesize	line size
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
ic	Add interval of confidence
title	title
fontfamily	Font family

Value

The function returns a graph for correlation

Author(s)

Gabriel Danilo Shimizu, <shimizu@uel.br>

Leandro Simoes Azeredo Goncalves

Examples

```
data("aristolochia")
with(aristolochia, correlation(trat,resp))
```

extract.model *Analysis: Extract models*

Description

This function allows extracting the model (type="model") or residuals (type="resids"). The model class depends on the function and can be (lm, drm or nls). This function also allows you to perform graphical analysis of residuals (type="residplot"), graphical analysis of standardized residuals (type="stdresidplot"), graph of theoretical quantiles (type="qqplot").

Usage

```
extract.model(model, type = "model")
```

Arguments

model	Object returned from an analysis function
type	output type

Value

Returns an object of class drm, lm or nls (type="model"), or vector of residuals (type="resids"), or graph of the residuals (type="residplot", type="stdresidplot", type=" qqplot").

Examples

```
data("aristolochia")
attach(aristolochia)
a=linear.linear(trat,resp,point = "mean")
extract.model(a,type = "qqplot")
```

gaussianreg *Analysis: Analogous to the Gaussian model/Bragg*

Description

Analysis: Analogous to the Gaussian model/Bragg

Usage

```
gaussianreg(
  trat,
  resp,
  npar = "g3",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  error = "SE",
  legend.position = "top",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	number of parameters (g3 or g4)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
error	Error bar (It can be SE - *default*, SD or FALSE)
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width

scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The model analogous to the three-parameter Gaussian is:

$$y = d \times e^{-b((x-e)^2)}$$

The model analogous to the three-parameter Gaussian is:

$$y = d \times c + (d - c) * e^{-b((x-e)^2)}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
gaussianreg(trat,resp)
```

GP *Analysis: Gompertz*

Description

The logistical models provide Gompertz modified logistical models. This model was extracted from the 'drc' package.

Usage

```
GP(
  trat,
  resp,
  npar = "g2",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  error = "SE",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.

npar	Number os parameters (g2, g3 or g4)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
error	Error bar (It can be SE - *default*, SD or FALSE)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The two-parameter Gompertz model is given by the function:

$$y = exp^{-exp^{b(x-e)}}$$

The three-parameter Gompertz model is given by the function:

$$y = d \times exp^{-exp^{b(x-e)}}$$

The four-parameter Gompertz model is given by the function:

$$y = c + (d - c)(exp^{-exp^{b(x-e)}})$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the drc package (Ritz et al., 2016)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley and Sons (p. 330).

Ritz, C.; Strebig, J.C. and Ritz, M.C. Package 'drc'. Creative Commons: Mountain View, CA, USA, 2016.

See Also

LL, CD, BC

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
GP(trat,resp, npar="g3")
```

---

granada *Dataset: Granada*

Description

The data are part of an experiment that studied the drying kinetics of pomegranate peel over time under an air-circulation oven. Mass loss was assessed.

Usage

```
data("granada")
```

Format

data.frame containing data set

`time` numeric vector with times

`WL` Numeric vector with response

Author(s)

Gabriel Danilo Shimizu

Examples

```
data(granada)
```

`hill` *Analysis: Hill*

Description

This function performs regression analysis using the Hill model.

Usage

```
hill(
  trat,
  resp,
  sample.curve = 1000,
  error = "SE",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  point = "all",
  width.bar = NA,
  r2 = "all",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.
`sample.curve`	Provide the number of observations to simulate curvature (default is 1000)
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`ylab`	Variable response name (Accepts the *expression*() function)
`xlab`	treatments name (Accepts the *expression*() function)
`theme`	ggplot2 theme (*default* is theme_bw())
`legend.position`	legend position (*default* is "top")
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`r2`	coefficient of determination of the mean or all values (*default* is all)
`textsize`	Font size
`pointsize`	shape size
`linesize`	line size
`linetype`	line type
`pointshape`	format point (default is 21)
`fillshape`	Fill shape
`colorline`	Color lines
`round`	round equation
`xname.formula`	Name of x in the equation
`yname.formula`	Name of y in the equation
`comment`	Add text after equation
`fontfamily`	Font family
`print.on`	Print output

Details

The Hill model is defined by:

$$y = \frac{a \times x^c}{b + x^c}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the aomisc package (Onofri, 2020)

Gabriel Danilo Shimizu

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Onofri A. (2020) The broken bridge between biologists and statisticians: a blog and R package, Statforbiology, IT, web: https://www.statforbiology.com

Examples

```
data("granada")
attach(granada)
hill(time,WL)
```

`interval.confidence` *Analysis: Interval of confidence*

Description

Interval of confidence in model regression

Usage

```
interval.confidence(model)
```

Arguments

`model`	Object analysis

Value

Return in the interval of confidence

Author(s)

Gabriel Danilo Shimizu

Examples

```
data("granada")
attach(granada)
a=LM(time, WL)
interval.confidence(a)
```

`linear.linear` *Analysis: Linear-Linear*

Description

This function performs linear linear regression analysis.

Usage

```
linear.linear(
  trat,
  resp,
  middle = 1,
  CI = FALSE,
  bootstrap.samples = 1000,
  sig.level = 0.05,
  error = "SE",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  point = "all",
  width.bar = NA,
  legend.position = "top",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.
`middle`	A scalar in [0,1]. This represents the range that the change-point can occur in. 0 means the change-point must occur at the middle of the range of x-values. 1 means that the change-point can occur anywhere along the range of the x-values.

CI	Whether or not a bootstrap confidence interval should be calculated. Defaults to FALSE because the interval takes a non-trivial amount of time to calculate
bootstrap.samples	The number of bootstrap samples to take when calculating the CI.
sig.level	What significance level to use for the confidence intervals.
error	Error bar (It can be SE - *default*, SD or FALSE)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
legend.position	legend position (*default* is "top")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The linear-linear model is defined by: First curve:

$$y = \beta_0 + \beta_1 \times x(x < breakpoint)$$

Second curve:

$$y = \beta_0 + \beta_1 \times breakpoint + w \times x(x > breakpoint)$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); breakpoint and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the SiZer package

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Chiu, G. S., R. Lockhart, and R. Routledge. 2006. Bent-cable regression theory and applications. Journal of the American Statistical Association 101:542-553.

Toms, J. D., and M. L. Lesperance. 2003. Piecewise regression: a tool for identifying ecological thresholds. Ecology 84:2034-2041.

See Also

quadratic.plateau, linear.plateau

Examples

```
library(AgroReg)
data("granada")
attach(granada)
linear.linear(time,WL)
```

---

linear.plateau *Analysis: Linear-Plateau*

Description

This function performs the linear-plateau regression analysis.

Usage

```
linear.plateau(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
```

```
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The linear-plateau model is defined by: First curve:

$$y = \beta_0 + \beta_1 \times x(x < breakpoint)$$

Second curve:

$$y = \beta_0 + \beta_1 \times breakpoint(x > breakpoint)$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); breakpoint and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Chiu, G. S., R. Lockhart, and R. Routledge. 2006. Bent-cable regression theory and applications. Journal of the American Statistical Association 101:542-553.

Toms, J. D., and M. L. Lesperance. 2003. Piecewise regression: a tool for identifying ecological thresholds. Ecology 84:2034-2041.

See Also

quadratic.plateau, linear.linear

Examples

```
library(AgroReg)
data("granada")
attach(granada)
linear.plateau(time,WL)
```

LL *Analysis: Log-logistic*

Description

Logistic models with three (LL.3), four (LL.4) or five (LL.5) continuous data parameters. This model was extracted from the drc package.

Usage

```
LL(
  trat,
  resp,
  npar = "LL.3",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	Number of model parameters
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)

ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The three-parameter log-logistic function with lower limit 0 is

$$y = 0 + \frac{d}{1 + \exp(b(\log(x) - \log(e)))}$$

The four-parameter log-logistic function is given by the expression

$$y = c + \frac{d - c}{1 + \exp(b(\log(x) - \log(e)))}$$

The function is symmetric about the inflection point (e).

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the drc package (Ritz et al., 2016)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Ritz, C.; Strebig, J.C.; Ritz, M.C. Package 'drc'. Creative Commons: Mountain View, CA, USA, 2016.

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
LL(trat,resp)
```

LM *Analysis: Linear, quadratic, quadratic inverse, cubic and quartic*

Description

Linear, quadratic, quadratic inverse, cubic and quartic regression.

Usage

```
LM(
  trat,
  resp,
  degree = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  error = "SE",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  point = "all",
  r2 = "all",
  theme = theme_classic(),
  legend.position = "top",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
```

```
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
degree	degree of the polynomial (0.5, 1, 2, 3 or 4)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Dependent variable name (Accepts the *expression*() function)
xlab	Independent variable name (Accepts the *expression*() function)
error	Error bar (It can be SE - *default*, SD or FALSE)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
point	defines whether you want to plot all points ("all") or only the mean ("mean")
r2	coefficient of determination of the mean or all values (*default* is all)
theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The linear model is defined by:

$$y = \beta_0 + \beta_1 \cdot x$$

The quadratic model is defined by:

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2$$

The quadratic inverse model is defined by:

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^{0.5}$$

The cubic model is defined by:

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$$

The quartic model is defined by:

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3 + \beta_4 \cdot x^4$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
LM(trat,resp, degree = 3)
```

LM13 *Analysis: Cubic without beta2*

Description

Degree 3 polynomial model without the beta 2 coefficient.

Usage

```
LM13(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  error = "SE",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.
`sample.curve`	Provide the number of observations to simulate curvature (default is 1000)
`ylab`	Dependent variable name (Accepts the *expression*() function)
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`xlab`	Independent variable name (Accepts the *expression*() function)
`theme`	ggplot2 theme (*default* is theme_classic())
`legend.position`	legend position (*default* is "top")
`r2`	coefficient of determination of the mean or all values (*default* is all)
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`scale`	Sets x scale (*default* is none, can be "log")
`textsize`	Font size

pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

Degree 3 polynomial model without the beta 2 coefficient is defined by:

$$y = \beta_0 + \beta_1 \cdot x + \beta_3 \cdot x^3$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("granada")
attach(granada)
LM13(time, WL)
```

LM13i *Analysis: Cubic inverse without beta2*

Description

Degree 3 polynomial inverse model without the beta 2 coefficient.

Usage

```
LM13i(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  error = "SE",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Dependent variable name (Accepts the *expression*() function)
error	Error bar (It can be SE - *default*, SD or FALSE)
xlab	Independent variable name (Accepts the *expression*() function)

theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

Inverse degree 3 polynomial model without the beta 2 coefficient is defined by:

$$y = \beta_0 + \beta_1 \cdot x + \beta_3 \cdot x^{1/3}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("granada")
attach(granada)
LM13i(time, WL)
```

LM23 *Analysis: Cubic without beta1*

Description

Degree 3 polynomial model without the beta 1 coefficient.

Usage

```
LM23(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  error = "SE",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Dependent variable name (Accepts the *expression*() function)
error	Error bar (It can be SE - *default*, SD or FALSE)
xlab	Independent variable name (Accepts the *expression*() function)

theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

Degree 3 polynomial model without the beta 2 coefficient is defined by:

$$y = \beta_0 + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("granada")
attach(granada)
LM23(time, WL)
```

LM23i *Analysis: Cubic inverse without beta1*

Description

Degree 3 polynomial inverse model without the beta 1 coefficient.

Usage

```
LM23i(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  error = "SE",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Dependent variable name (Accepts the *expression*() function)
error	Error bar (It can be SE - *default*, SD or FALSE)
xlab	Independent variable name (Accepts the *expression*() function)

theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

Inverse degree 3 polynomial model without the beta 1 coefficient is defined by:

$$y = \beta_0 + \beta_2 \cdot x^{1/2} + \beta_3 \cdot x^{1/3}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("granada")
attach(granada)
LM23i(time, WL)
```

LM2i3 *Analysis: Cubic without beta1, with inverse beta3*

Description

Degree 3 polynomial model without the beta 1 coefficient, with inverse beta3.

Usage

```
LM2i3(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  error = "SE",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Dependent variable name (Accepts the *expression*() function)
error	Error bar (It can be SE - *default*, SD or FALSE)
xlab	Independent variable name (Accepts the *expression*() function)

theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

Inverse degree 3 polynomial model without the beta 2 coefficient is defined by:

$$y = \beta_0 + \beta_1 \cdot x^2 + \beta_3 \cdot x^{1/3}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("granada")
attach(granada)
LM2i3(time, WL)
```

LM_i *Analysis: Linear, quadratic, quadratic inverse, cubic and quartic without intercept*

Description

Linear, quadratic, quadratic inverse, cubic and quartic regression.

Usage

```
LM_i(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  error = "SE",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  xlab = "Independent",
  degree = NA,
  theme = theme_classic(),
  legend.position = "top",
  point = "all",
  r2 = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.

sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Dependent variable name (Accepts the *expression*() function)
error	Error bar (It can be SE - *default*, SD or FALSE)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
xlab	Independent variable name (Accepts the *expression*() function)
degree	degree of the polynomial (0.5, 1, 2, 3 or 4)
theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
point	defines whether you want to plot all points ("all") or only the mean ("mean")
r2	coefficient of determination of the mean or all values (*default* is all)
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The linear model is defined by:

$$y = \beta_1 \cdot x$$

The quadratic model is defined by:

$$y = \beta_1 \cdot x + \beta_2 \cdot x^2$$

The quadratic inverse model is defined by:

$$y = \beta_1 \cdot x + \beta_2 \cdot x^{0.5}$$

The cubic model is defined by:

$$y = \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$$

The quartic model is defined by:

$$y = \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3 + \beta_4 \cdot x^4$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
LM_i(trat,resp, degree = 3)
```

---

loessreg *Analysis: loess regression (degree 0, 1 or 2)*

Description

Fit a polynomial surface determined by one or more numerical predictors, using local fitting.

Usage

```
loessreg(
  trat,
  resp,
  degree = 2,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  fontfamily = "sans",
```

```
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.
`degree`	Degree polynomial (0,1 or 2)
`sample.curve`	Provide the number of observations to simulate curvature (default is 1000)
`ylab`	Variable response name (Accepts the *expression*() function)
`xlab`	treatments name (Accepts the *expression*() function)
`theme`	ggplot2 theme (*default* is theme_bw())
`legend.position`	legend position (*default* is c(0.3,0.8))
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`scale`	Sets x scale (*default* is none, can be "log")
`textsize`	Font size
`pointsize`	shape size
`linesize`	line size
`linetype`	line type
`pointshape`	format point (default is 21)
`fillshape`	Fill shape
`colorline`	Color lines
`fontfamily`	Font family
`print.on`	Print output

Value

The function returns a list containing the loess regression and graph using ggplot2.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

See Also

loess

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
loessreg(trat,resp)
```

LOG *Analysis: Logarithmic*

Description

This function performs logarithmic regression analysis.

Usage

```
LOG(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.

sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is c(0.3,0.8))
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The logarithmic model is defined by:

$$y = \beta_0 + \beta_1 ln(\cdot x)$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
library(AgroReg)
resp=c(10,8,6.8,6,5,4.3,4.1,4.2,4.1)
trat=seq(1,9,1)
LOG(trat,resp)
```

LOG2 *Analysis: Logarithmic quadratic*

Description

This function performs logarithmic quadratic regression analysis.

Usage

```
LOG2(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is c(0.3,0.8))
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The logarithmic model is defined by:

$$y = \beta_0 + \beta_1 ln(\cdot x) + \beta_2 ln(\cdot x)^2$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
library(AgroReg)
resp=c(10,8,6.8,6,5,4.3,4.1,4.2,4.1)
trat=seq(1,9,1)
LOG2(trat,resp)
```

---

logistic *Analysis: Logistic*

Description

Logistic models with three (L.3), four (L.4) or five (L.5) continuous data parameters. This model was extracted from the drc package.

Usage

```
logistic(
  trat,
  resp,
  npar = "L.3",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
```

```
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	Number of model parameters
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The three-parameter logistic function with lower limit 0 is

$$y = 0 + \frac{d}{1 + \exp(b(x - e))}$$

The four-parameter logistic function is given by the expression

$$y = c + \frac{d - c}{1 + \exp(b(x - e))}$$

The five-parameter logistic function is given by the expression

$$y = c + \frac{d - c}{1 + \exp(b(x - e))^f}$$

The function is symmetric about the inflection point (e).

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the drc package (Ritz et al., 2016)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Ritz, C.; Strebig, J.C.; Ritz, M.C. Package 'drc'. Creative Commons: Mountain View, CA, USA, 2016.

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
logistic(trat,resp)
```

lorentz *Analysis: Lorentz*

Description

Analysis: Lorentz

Usage

```
lorentz(
  trat,
  resp,
  npar = "lo3",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  error = "SE",
  legend.position = "top",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	number of parameters (lo3 or lo4)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)

xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
error	Error bar (It can be SE - *default*, SD or FALSE)
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The model to the three-parameter Lorentz is:

$$y = fracd1 + b(x - e)^2$$

The model to the three-parameter Lorentz is:

$$y = c + fracd - c1 + b(x - e)^2$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the aomisc package (Onofri, 2020)

Gabriel Danilo Shimizu

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Onofri A. (2020) The broken bridge between biologists and statisticians: a blog and R package, Statforbiology, IT, web: https://www.statforbiology.com

Examples

```
library(AgroReg)
data("granada")
attach(granada)
x=time[length(time):1]
lorentz(x,WL)
```

`midilli` *Analysis: Midilli*

Description

This function performs Midilli regression analysis.

Usage

```
midilli(
  trat,
  resp,
  initial = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
```

```
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
initial	List starting estimates
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The exponential model is defined by:

$$y = \alpha \times e^{-\beta \cdot x^n} + \theta \cdot x$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
library(AgroReg)
data("granada")
attach(granada)
midilli(time,100-WL)
```

`midillim` *Analysis: Modified Midilli*

Description

This function performs modified Midilli regression analysis.

Usage

```
midillim(
  trat,
  resp,
  initial = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
```

```
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
initial	List starting estimates
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The exponential model is defined by:

$$y = \alpha \times e^{-\beta \cdot x} + \theta \cdot x$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
library(AgroReg)
data("granada")
attach(granada)
midillim(time,100-WL)
```

`mitscherlich` *Analysis: Mitscherlich*

Description

This function performs Mitscherlich regression analysis.

Usage

```
mitscherlich(
  trat,
  resp,
  initial = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
```

```
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
initial	List Initial parameters (A, b, e)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation

yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The Mitscherlich model is defined by:

$$y = A \times (1 - 10^{-eb-ex})$$

where "y" is the yield obtained when "b" units of a nutrient are in the soil and "x" units of it are added as fertilizer, "A" is the maximum yield, and "e" is the proportionality factor, has recently received increasing interest.

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("granada")
attach(granada)
mitscherlich(time,WL)
```

MM *Analysis: Michaelis-Menten*

Description

This function performs regression analysis using the Michaelis-Menten model.

Usage

```
MM(
  trat,
  resp,
  npar = "mm2",
  sample.curve = 1000,
  error = "SE",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  point = "all",
  width.bar = NA,
  r2 = "all",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	Number of parameters (mm2 or mm3)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
error	Error bar (It can be SE - *default*, SD or FALSE)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
point	defines whether you want to plot all points ("all") or only the mean ("mean")

width.bar	Bar width
r2	coefficient of determination of the mean or all values (*default* is all)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The two-parameter Michaelis-Menten model is defined by:

$$y = \frac{Vm \times x}{k + x}$$

The three-parameter Michaelis-Menten model is defined by:

$$y = c + \frac{Vm \times x}{k + x}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
data("granada")
attach(granada)
MM(time,WL)
MM(time,WL,npar="mm3")
```

`newton` *Analysis: Newton*

Description

This function performs exponential regression analysis. This model was used by Newton.

Usage

```
newton(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.

`sample.curve`	Provide the number of observations to simulate curvature (default is 1000)
`ylab`	Variable response name (Accepts the *expression*() function)
`xlab`	treatments name (Accepts the *expression*() function)
`theme`	ggplot2 theme (*default* is theme_bw())
`legend.position`	legend position (*default* is "top")
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`r2`	coefficient of determination of the mean or all values (*default* is all)
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`scale`	Sets x scale (*default* is none, can be "log")
`textsize`	Font size
`pointsize`	shape size
`linesize`	line size
`linetype`	line type
`pointshape`	format point (default is 21)
`fillshape`	Fill shape
`colorline`	Color lines
`round`	round equation
`yname.formula`	Name of y in the equation
`xname.formula`	Name of x in the equation
`comment`	Add text after equation
`fontfamily`	Font family
`print.on`	Print output

Details

The exponential model is defined by:

$$y = e^{-\beta \cdot x} \cdot x$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Siqueira, V. C., Resende, O., and Chaves, T. H. (2013). Mathematical modelling of the drying of jatropha fruit: an empirical comparison. Revista Ciencia Agronomica, 44, 278-285.

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
newton(trat,resp+0.001)
```

`Nreg` *Analysis: Graph for not significant trend*

Description

Graph for non-significant trend. Can be used within the multicurve command

Usage

```
Nreg(
  trat,
  resp,
  ylab = "Dependent",
  xlab = "Independent",
  error = "SE",
  theme = theme_classic(),
  legend.position = "top",
  legend.text = "not~significant",
  legend.add.mean = TRUE,
  legend.add.mean.name = "hat(y)",
  width.bar = NA,
  point = "all",
  textsize = 12,
  add.line = FALSE,
  add.line.mean = FALSE,
  linesize = 0.8,
  linetype = 1,
  pointsize = 4.5,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
ylab	Dependent variable name (Accepts the *expression*() function)
xlab	Independent variable name (Accepts the *expression*() function)
error	Error bar (It can be SE - *default*, SD or FALSE)
theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
legend.text	legend text
legend.add.mean	Add average in legend
legend.add.mean.name	Add media name
width.bar	Bar width
point	defines whether you want to plot all points ("all") or only the mean ("mean")
textsize	Font size
add.line	Add line
add.line.mean	Add line mean
linesize	line size
linetype	line type
pointsize	shape size
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
fontfamily	Font family
print.on	Print output

Value

The function returns an exploratory graph of segments

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
Nreg(trat,resp)
```

PAGE *Analysis: Page*

Description

This function performs exponential page regression analysis.

Usage

```
PAGE(
  trat,
  resp,
  initial = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
initial	Starting estimates
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)

xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The exponential model is defined by:

$$y = e^{-k \cdot x^n}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
library(AgroReg)
data("granada")
attach(granada)
PAGE(time,100-WL)
```

---

peleg *Analysis: Peleg*

Description

This function performs Peleg regression analysis.

Usage

```
peleg(
  trat,
  resp,
  initial = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
initial	Starting estimates
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The Peleg model is defined by:

$$y = \frac{(1-x)}{a+bx}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Examples

```
library(AgroReg)
data("granada")
attach(granada)
peleg(time,WL)
```

---

plateau.linear *Analysis: Plateau-Linear*

Description

This function performs the plateau-linear regression analysis.

Usage

```
plateau.linear(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  xname.formula = "x",
  yname.formula = "y",
```

```
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The plateau-linear model is defined by: First curve:

$$y = \beta_0 + \beta_1 \times breakpoint(x < breakpoint)$$

Second curve:

$$y = \beta_0 + \beta_1 \times x(x > breakpoint)$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); breakpoint and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Chiu, G. S., R. Lockhart, and R. Routledge. 2006. Bent-cable regression theory and applications. Journal of the American Statistical Association 101:542-553.

Toms, J. D., and M. L. Lesperance. 2003. Piecewise regression: a tool for identifying ecological thresholds. Ecology 84:2034-2041.

See Also

quadratic.plateau, linear.linear

Examples

```
library(AgroReg)
data("granada")
attach(granada)
x=time[length(time):1]
plateau.linear(x,WL)
```

`plateau.quadratic` *Analysis: Plateau-quadratic*

Description

This function performs the plateau-quadratic regression analysis.

Usage

```
plateau.quadratic(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
```

```
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)

plquadratic(x, a, breakpoint, b, c)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines

round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output
x	Numeric vector with dependent variable.
a	The plateau value
breakpoint	breakpoint value
b	Linear term
c	Quadratic term

Details

The Plateau-quadratic model is defined by:

First curve:

$$y = \beta_0 + \beta_1 \cdot breakpoint + \beta_2 \cdot breakpoint^2 (x < breakpoint)$$

Second curve:

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 (x > breakpoint)$$

or

$$y = a + b(x + breakpoint) + c(x + breakpoint)^2 (x > breakpoint)$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Miguez, F. (2020). nlraa: nonlinear Regression for Agricultural Applications. R package version 0.65.

Chiu, G. S., R. Lockhart, and R. Routledge. 2006. Bent-cable regression theory and applications. Journal of the American Statistical Association 101:542-553.

Toms, J. D., and M. L. Lesperance. 2003. Piecewise regression: a tool for identifying ecological thresholds. Ecology 84:2034-2041.

See Also

linear.linear, linear.plateau

Examples

```
library(AgroReg)
data("granada")
attach(granada)
x=time[length(time):1]
plateau.quadratic(x,WL)
```

`plot_arrange` *Merge multiple curves into a single graph*

Description

Merge multiple curves into a single graph

Usage

```
plot_arrange(
  plots,
  point = "mean",
  theme = theme_classic(),
  legend.title = NULL,
  legend.position = "top",
  trat = NA,
  gray = FALSE,
  ylab = "Dependent",
  xlab = "Independent",
  widthbar = 0,
  pointsize = 4.5,
  linesize = 0.8,
  textsize = 12,
  legendsize = 12,
  legendtitlesize = 12,
  fontfamily = "sans"
)
```

Arguments

`plots`	list with objects of type analysis.
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`theme`	ggplot2 theme (*default* is theme_classic())
`legend.title`	caption title

legend.position	legend position (*default* is c(0.3,0.8))
trat	name of the curves
gray	gray scale (*default* is FALSE)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
widthbar	bar width (*default* is 0.3)
pointsize	shape size
linesize	line size
textsize	Font size
legendsize	Legend size text
legendtitlesize	Title legend size
fontfamily	font family

Value

The function returns a graph joining the outputs of the functions LM_model, LL_model, BC_model, CD_model, loess_model, normal_model, piecewise_model and N_model

Author(s)

Gabriel Danilo Shimizu

Examples

```
library(AgroReg)
library(ggplot2)
data("aristolochia")
attach(aristolochia)
a=LM(trat,resp)
b=LL(trat,resp,npar = "LL.3")
plot_arrange(list(a,b))

models <- c("LM1", "LL3")
r <- lapply(models, function(x) {
r <- with(granada, regression(time, WL, model = x,print.on=FALSE))
})
plot_arrange(r,trat=models,ylab="WL (%)",xlab="Time (Minutes)")

models = c("asymptotic_neg", "biexponential", "LL4", "BC4", "CD5", "linear.linear",
           "linear.plateau", "quadratic.plateau", "mitscherlich", "MM2")
m = lapply(models, function(x) {
           m = with(granada, regression(time, WL, model = x,print.on=FALSE))})
           plot_arrange(m, trat = paste("(",models,")"))
```

potential *Analysis: Potencial*

Description

This function performs potencial regression analysis.

Usage

```
potential(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())

`legend.position`	legend position (*default* is "top")
`error`	Error bar (It can be SE - *default*, SD or FALSE)
`r2`	coefficient of determination of the mean or all values (*default* is all)
`point`	defines whether you want to plot all points ("all") or only the mean ("mean")
`width.bar`	Bar width
`scale`	Sets x scale (*default* is none, can be "log")
`textsize`	Font size
`pointsize`	shape size
`linesize`	line size
`linetype`	line type
`pointshape`	format point (default is 21)
`fillshape`	Fill shape
`colorline`	Color lines
`round`	round equation
`yname.formula`	Name of y in the equation
`xname.formula`	Name of x in the equation
`comment`	Add text after equation
`fontfamily`	Font family
`print.on`	Print output

Details

The exponential model is defined by:

$$y = \alpha \times trat^{\beta}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Siqueira, V. C., Resende, O., & Chaves, T. H. (2013). Mathematical modelling of the drying of jatropha fruit: an empirical comparison. Revista Ciencia Agronomica, 44, 278-285.

Examples

```
library(AgroReg)
data("granada")
attach(granada)
potential(time,WL)
```

`quadratic.plateau` *Analysis: Quadratic-plateau*

Description

This function performs the quadratic-plateau regression analysis.

Usage

```
quadratic.plateau(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.

sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The quadratic-plateau model is defined by:

First curve:

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 (x < breakpoint)$$

Second curve:

$$y = \beta_0 + \beta_1 \cdot breakpoint + \beta_2 \cdot breakpoint^2 (x > breakpoint)$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Chiu, G. S., R. Lockhart, and R. Routledge. 2006. Bent-cable regression theory and applications. Journal of the American Statistical Association 101:542-553.

Toms, J. D., and M. L. Lesperance. 2003. Piecewise regression: a tool for identifying ecological thresholds. Ecology 84:2034-2041.

See Also

linear.linear, linear.plateau

Examples

```
library(AgroReg)
data("granada")
attach(granada)
quadratic.plateau(time,WL)
```

---

regression *Analysis: Regression linear or nonlinear*

Description

This function is a simplification of all the analysis functions present in the package.

Usage

```
regression(
  trat,
  resp,
  model = "LM1",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  point = "all",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  pointshape = 21,
  round = NA,
  fontfamily = "sans",
  error = "SE",
```

```
  width.bar = NA,
  xname.formula = "x",
  yname.formula = "y",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
model	model regression (*default* is LM1)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is c(0.3,0.8))
point	defines whether you want to plot all points ("all") or only the mean ("mean")
textsize	Font size
pointsize	shape size
linesize	line size
pointshape	format point (default is 21)
round	round equation
fontfamily	Font family
error	Error bar (It can be SE - *default*, SD or FALSE)
width.bar	Bar width
xname.formula	Name of x in the equation
yname.formula	Name of y in the equation
print.on	Print output

Details

To change the regression model, change the "model" argument to:

1. **N:** Graph for not significant trend.
2. **loess0:** Loess non-parametric degree 0
3. **loess1:** Loess non-parametric degree 1
4. **loess2:** Loess non-parametric degree 2
5. **LM0.5:** Quadratic inverse
6. **LM1:** Linear regression.
7. **LM2:** Quadratic
8. **LM3:** Cubic

9. **LM4:** Quartic
10. **LM0.5_i:** Quadratic inverse without intercept.
11. **LM1_i:** Linear without intercept.
12. **LM2_i:** Quadratic regression without intercept.
13. **LM3_i:** Cubic without intercept.
14. **LM4_i:** Quartic without intercept.
15. **LM13:** Cubic without beta2
16. **LM13i:** Cubic inverse without beta2
17. **LM23:** Cubic without beta1
18. **LM23i:** Cubic inverse without beta2
19. **LM2i3:** Cubic without beta1, with inverse beta3
20. **valcam:** Valcam
21. **L3:** Three-parameter logistics.
22. **L4:** Four-parameter logistics.
23. **L5:** Five-parameter logistics.
24. **LL3:** Three-parameter log-logistics.
25. **LL4:** Four-parameter log-logistics.
26. **LL5:** Five-parameter log-logistics.
27. **BC4:** Brain-Cousens with four parameter.
28. **BC5:** Brain-Cousens with five parameter.
29. **CD4:** Cedergreen-Ritz-Streibig with four parameter.
30. **CD5:** Cedergreen-Ritz-Streibig with five parameter.
31. **weibull3:** Weibull with three parameter.
32. **weibull4:** Weibull with four parameter.
33. **GP2:** Gompertz with two parameter.
34. **GP3:** Gompertz with three parameter.
35. **GP4:** Gompertz with four parameter.
36. **VB:** Von Bertalanffy
37. **lo3:** Lorentz with three parameter
38. **lo4:** Lorentz with four parameter
39. **beta:** Beta
40. **gaussian3:** Analogous to the Gaussian model/Bragg with three parameters.
41. **gaussian4:** Analogous to the Gaussian model/Bragg with four parameters.
42. **linear.linear:** Linear-linear
43. **linear.plateau:** Linear-plateau
44. **quadratic.plateau:** Quadratic-plateau
45. **plateau.linear:** Plateau-linear

46. **plateau.quadratic:** Plateau-Quadratic
47. **log:** Logarithmic
48. **log2:** Logarithmic quadratic
49. **thompson:** Thompson
50. **asymptotic:** Exponential
51. **asymptotic_neg:** Exponential negative
52. **asymptotic_i:** Exponential without intercept.
53. **asymptotic_ineg:** Exponential negative without intercept.
54. **biexponential:** Biexponential
55. **mitscherlich:** Mitscherlich
56. **yieldloss:** Yield-loss
57. **hill:** Hill
58. **MM2:** Michaelis-Menten with two parameter.
59. **MM3:** Michaelis-Menten with three parameter.
60. **SH:** Steinhart-Hart
61. **page:** Page
62. **newton:** Newton
63. **potential:** Potential
64. **midilli:** Midilli
65. **midillim:** Modified Midilli
66. **AM:** Avhad and Marchetti
67. **peleg:** Peleg
68. **VG:** Vega-Galvez

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
regression(trat, resp)
```

SH *Analysis: Steinhart-Hart*

Description

The Steinhart-Hart model. The Steinhart-Hart equation is a model used to explain the behavior of a semiconductor at different temperatures, however, Zhai et al. (2020) used this model to relate plant density and grain yield.

Usage

```
SH(
  trat,
  resp,
  initial = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  error = "SE",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
initial	Starting estimates
sample.curve	Provide the number of observations to simulate curvature (default is 1000)

ylab	Variable response name (Accepts the *expression*() function)
xlab	Treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	Legend position (*default* is "top")
r2	Coefficient of determination of the mean or all values (*default* is all)
error	Error bar (It can be SE - *default*, SD or FALSE)
point	Defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	Shape size
linesize	Line size
linetype	line type
pointshape	Format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The model function for the Steinhart-Hart model is:

$$y = \frac{1}{A + B \times ln(x) + C \times ln(x)^3}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Zhai, L., Li, H., Song, S., Zhai, L., Ming, B., Li, S., ... & Zhang, L. (2021). Intra-specific competition affects the density tolerance and grain yield of maize hybrids. Agronomy Journal, 113(1), 224-23. doi:10.1002/agj2.20438

See Also

LL, CD,GP

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
SH(trat,resp)
```

`stat_param` *Analysis: Other statistical parameters*

Description

This function calculates other statistical parameters such as Mean (Bias) Error, Relative Mean (Bias) Error, Mean Absolute Error, Relative Mean Absolute Error, Root Mean Square Error, Relative Root Mean Square Error, Modeling Efficiency, Standard deviation of differences, Coefficient of Residual Mass.

Usage

```
stat_param(models, names_model = NA, round = 3)
```

Arguments

`models`	List with objects of type analysis
`names_model`	Names of the models
`round`	Round numbers

Value

Returns a table with the statistical parameters for choosing the model.

Author(s)

Gabriel Danilo Shimizu

Examples

```
library(AgroReg)
data(granada)
attach(granada)
a=LM(time,WL)
b=LL(time,WL)
c=BC(time,WL)
d=weibull(time,WL)
stat_param(models=list(a,b,c,d))
```

---

thompson *Analysis: Thompson*

Description

This function performs Thompson regression analysis.

Usage

```
thompson(
  trat,
  resp,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  error = "SE",
  r2 = "all",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is c(0.3,0.8))
error	Error bar (It can be SE - *default*, SD or FALSE)
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The logarithmic model is defined by:

$$y = \beta_1 ln(\cdot x) + \beta_2 ln(\cdot x)^2$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Sadeghi, E., Haghighi Asl, A., & Movagharnejad, K. (2019). Mathematical modelling of infrared-dried kiwifruit slices under natural and forced convection. Food science & nutrition, 7(11), 3589-3606.

Examples

```
library(AgroReg)
resp=c(10,8,6.8,6,5,4.3,4.1,4.2,4.1)
trat=seq(1,9,1)
thompson(trat,resp)
```

`valcam` *Analysis: Valcam*

Description

This function performs Valcam regression analysis.

Usage

```
valcam(
  trat,
  resp,
  sample.curve = 1000,
  error = "SE",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "mean",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
```

```
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
error	Error bar (It can be SE - *default*, SD or FALSE)
ylab	Dependent variable name (Accepts the *expression*() function)
xlab	Independent variable name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The Valcam model is defined by:

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^1.5 + \beta_3 \cdot x^2$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Siqueira, V. C., Resende, O., & Chaves, T. H. (2013). Mathematical modelling of the drying of jatropha fruit: an empirical comparison. Revista Ciencia Agronomica, 44, 278-285.

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
valcam(trat,resp)
```

VB *Analysis: Von Bertalanffy*

Description

The Von Bertalanffy model. It's a kind of growth curve for a time series and takes its name from its creator, Ludwig von Bertalanffy. It is a special case of the generalized logistic function. The growth curve (biology) is used to model the average length from age in animals.

Usage

```
VB(
  trat,
  resp,
  initial = NA,
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "all",
  error = "SE",
  point = "all",
  width.bar = NA,
  scale = "none",
```

```
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
initial	Starting estimates
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	Treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	Legend position (*default* is "top")
r2	Coefficient of determination of the mean or all values (*default* is all)
error	Error bar (It can be SE - *default*, SD or FALSE)
point	Defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	Shape size
linesize	Line size
linetype	line type
pointshape	Format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The model function for the von Bertalanffy model is:

$$y = L(1 - exp(-k(t - t0)))$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

Examples

```
library(AgroReg)
x=seq(1,20)
y=c(0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 0.91,
    0.92, 0.94, 0.96, 0.98, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00)
VB(x,y)
```

VG *Analysis: Vega-Galvez*

Description

This function performs Vega-Galvez regression analysis.

Usage

```
VG(
  trat,
  resp,
  sample.curve = 1000,
  error = "SE",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  r2 = "mean",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
```

```
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
error	Error bar (It can be SE - *default*, SD or FALSE)
ylab	Dependent variable name (Accepts the *expression*() function)
xlab	Independent variable name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_classic())
legend.position	legend position (*default* is "top")
r2	coefficient of determination of the mean or all values (*default* is all)
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The Vega-Galvez model is defined by:

$$y = \beta_0 + \beta_1(\sqrt{x})$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Sadeghi, E., Haghighi Asl, A., and Movagharnejad, K. (2019). Mathematical modelling of infrared-dried kiwifruit slices under natural and forced convection. Food science & nutrition, 7(11), 3589-3606.

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
VG(trat,resp)
```

---

weibull *Analysis: Weibull*

Description

The w3' and 'w4' logistical models provide Weibull. This model was extracted from the 'drc' package.

Usage

```
weibull(
  trat,
  resp,
  npar = "w3",
  sample.curve = 1000,
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
```

```
  legend.position = "top",
  r2 = "all",
  ic = FALSE,
  fill.ic = "gray70",
  alpha.ic = 0.5,
  error = "SE",
  point = "all",
  width.bar = NA,
  scale = "none",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

trat	Numeric vector with dependent variable.
resp	Numeric vector with independent variable.
npar	Number of model parameters (*default* is w3)
sample.curve	Provide the number of observations to simulate curvature (default is 1000)
ylab	Variable response name (Accepts the *expression*() function)
xlab	Treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	Legend position (*default* is "top")
r2	Coefficient of determination of the mean or all values (*default* is all)
ic	Add interval of confidence
fill.ic	Color interval of confidence
alpha.ic	confidence interval transparency level
error	Error bar (It can be SE - *default*, SD or FALSE)
point	Defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
scale	Sets x scale (*default* is none, can be "log")
textsize	Font size

pointsize	Shape size
linesize	Line size
linetype	line type
pointshape	Format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
fontfamily	Font family
print.on	Print output

Details

The three-parameter Weibull model is given by the expression

$$y = d\exp(-\exp(b(\log(x) - e)))$$

Fixing the lower limit at 0 yields the four-parameter model

$$y = c + (d - c)(1 - \exp(-\exp(b(\log(x) - \log(e)))))$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the drc package (Ritz et al., 2016)

Gabriel Danilo Shimizu

Leandro Simoes Azeredo Goncalves

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Ritz, C.; Strebig, J.C. and Ritz, M.C. Package 'drc'. Creative Commons: Mountain View, CA, USA, 2016.

See Also

LL, CD,GP

Examples

```
library(AgroReg)
data("aristolochia")
attach(aristolochia)
weibull(trat,resp)
```

`yieldloss` *Analysis: Yield-loss*

Description

This function performs regression analysis using the Yield loss model.

Usage

```
yieldloss(
  trat,
  resp,
  sample.curve = 1000,
  error = "SE",
  ylab = "Dependent",
  xlab = "Independent",
  theme = theme_classic(),
  legend.position = "top",
  point = "all",
  width.bar = NA,
  r2 = "all",
  textsize = 12,
  pointsize = 4.5,
  linesize = 0.8,
  linetype = 1,
  pointshape = 21,
  fillshape = "gray",
  colorline = "black",
  round = NA,
  yname.formula = "y",
  xname.formula = "x",
  comment = NA,
  scale = "none",
  fontfamily = "sans",
  print.on = TRUE
)
```

Arguments

`trat`	Numeric vector with dependent variable.
`resp`	Numeric vector with independent variable.

sample.curve	Provide the number of observations to simulate curvature (default is 1000)
error	Error bar (It can be SE - *default*, SD or FALSE)
ylab	Variable response name (Accepts the *expression*() function)
xlab	treatments name (Accepts the *expression*() function)
theme	ggplot2 theme (*default* is theme_bw())
legend.position	legend position (*default* is "top")
point	defines whether you want to plot all points ("all") or only the mean ("mean")
width.bar	Bar width
r2	coefficient of determination of the mean or all values (*default* is all)
textsize	Font size
pointsize	shape size
linesize	line size
linetype	line type
pointshape	format point (default is 21)
fillshape	Fill shape
colorline	Color lines
round	round equation
yname.formula	Name of y in the equation
xname.formula	Name of x in the equation
comment	Add text after equation
scale	Sets x scale (*default* is none, can be "log")
fontfamily	Font family
print.on	Print output

Details

The Yield Loss model is defined by:

$$y = \frac{i \times x}{1 + \frac{i}{A} \times x}$$

Value

The function returns a list containing the coefficients and their respective values of p; statistical parameters such as AIC, BIC, pseudo-R2, RMSE (root mean square error); largest and smallest estimated value and the graph using ggplot2 with the equation automatically.

Author(s)

Model imported from the aomisc package (Onofri, 2020)

Gabriel Danilo Shimizu

References

Seber, G. A. F. and Wild, C. J (1989) Nonlinear Regression, New York: Wiley & Sons (p. 330).

Onofri A. (2020) The broken bridge between biologists and statisticians: a blog and R package, Statforbiology, IT, web: https://www.statforbiology.com

Examples

```
data("granada")
attach(granada)
yieldloss(time,WL)
```

Index