Package 'sars'

December 14, 2022

Type Package

Title Fit and Compare Species-Area Relationship Models Using Multimodel Inference

Version 1.3.6

Description Implements the basic elements of the multi-model inference paradigm for up to twenty species-area relationship models (SAR), using simple R list-objects and functions, as in Triantis et al. 2012 <DOI:10.1111/j.1365-2699.2011.02652.x>. The package is scalable and users can easily create their own model and data objects. Additional SAR related functions are provided.

License GPL-3 | file LICENSE

URL https://github.com/txm676/sars, https://txm676.github.io/sars/

BugReports https://github.com/txm676/sars/issues

Imports graphics, nortest, stats, utils, crayon, cli, numDeriv, doParallel, foreach, parallel, AICcmodavg

Depends R(>= 3.6.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

NeedsCompilation no

Author Thomas J. Matthews [aut, cre] (<https://orcid.org/0000-0002-7624-244X>), Francois Guilhaumon [aut] (<https://orcid.org/0000-0003-4707-8932>), Kevin Cazelles [rev] (<https://orcid.org/0000-0001-6619-9874>)

Maintainer Thomas J. Matthews <txm676@gmail.com>

Repository CRAN

Date/Publication 2022-12-14 11:20:03 UTC

R topics documented:

sars-package *sars: Fit and compare species-area relationship models using multi-model inference*

Description

This package provides functions to fit twenty models to species-area relationship (SAR) data (see Triantis et al. 2012), plot the model fits, and to construct a multimodel SAR curve using information criterion weights. A number of additional SAR functions are provided, e.g. to fit the log-log power model, the general dynamic model of island biogeography (GDM), Coleman's Random Placement model, and piecewise ISAR models (i.e. models with thresholds in the ISAR).

Details

Functions are provided to fit 20 individual SAR models. Nineteen are fitted using non-linear regression, whilst a single model (the linear model) is fitted using linear regression. Each model has its own function (e.g. `sar_power`). A set of multiple model fits can be combined into a fit collection (`sar_multi`). Plotting functions (`plot.sars`) are provided that enable individual model fits to be plotted on their own, or the fits of multiple models to be overlayed on the same plot. Model fits can be validated using a number of checks, e.g. the normality and homogeneity of the model residuals can be assessed.

A multimodel SAR curve can be constructed using the `sar_average` function. This fits up to twenty SAR models and constructs the multimodel curve (with confidence intervals) using information criterion weights (see `summary.sars` to calculate a table of models ranked by information criterion weight). The `plot.multi` functions enables the multimodel SAR curve to be plotted with or without the fits of the individual models.

Other SAR related functions include: (i) `lin_pow`, which fits the log-log power model and enables comparison of the model parameters with those calculated using the non-linear power model, (ii) `gdm`, which fits the general dynamic model of island biogeography (Whittaker et al. 2008) using several different functions, and (iii) `coleman`, which fits Coleman's (1981) random placement model to a species-site abundance matrix. Version 1.3.0 has added functions for fitting, evaluating and plotting a range of commonly used piecewise SAR models (`sar_threshold`).

Author(s)

Thomas J. Matthews and Francois Guilhaumon

References

Coleman, B. D. (1981). On random placement and species-area relations. Mathematical Biosciences, 54, 191-215.

Guilhaumon, F., Mouillot, D., & Gimenez, O. (2010). mmSAR: an R-package for multimodel species–area relationship inference. Ecography, 33, 420-424.

Matthews, T.J., Guilhaumon, F., Triantis, K.A, Borregaard, M.K., & Whittaker, R.J. (2015b) On the form of species–area relationships in habitat islands and true islands. Global Ecology & Biogeography. DOI: 10.1111/geb.12269.

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species–area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Whittaker, R.J., Triantis, K.A. & Ladle, R.J. (2008) A general dynamic theory of oceanic island biogeography. Journal of Biogeography, 35, 977-994.

See Also

https://github.com/txm676/sars

Examples

```
data(galap, package = "sars")
#fit the power model
fit <- sar_power(galap)
summary(fit)
plot(fit)

#Construct a multimodel averaged SAR curve, using no grid_start simply
#for speed (not recommended - see documentation for sar_average())
fit_multi <- sar_average(data = galap, grid_start = "none")
summary(fit_multi)
plot(fit_multi)
```

`aegean` *A SAR dataset describing invertebrates on islands in the Aegean Sea, Greece*

Description

A sample dataset in the correct sars format: contains the areas of a number of islands in the Aegean Sea, Greece, and the number of invertebrate species recorded on each island.

Usage

```
data(aegean)
```

Format

A data frame with 2 columns and 90 rows. Each row contains the area of an island in the Aegean (1st column) and the number of inverts on that island (2nd column).

Source

Sfenthourakis, S. & Triantis K.A. (2009). Habitat diversity, ecological requirements of species and the Small Island Effect. Diversity Distrib.,15, 131–140.

Examples

```
data(aegean)
```

`aegean2` *A SAR dataset describing plants on islands in the Aegean Sea, Greece*

Description

A sample dataset in the correct sars format: contains the areas of a number of islands in the Aegean Sea, Greece, and the number of plant species recorded on each island.

Usage

```
data(aegean2)
```

Format

A data frame with 2 columns and 173 rows. Each row contains the area of an island in the Aegean (1st column) and the number of plants on that island (2nd column).

Source

Matthews, T.J. et al. (In review) Unravelling the small-island effect through phylogenetic community ecology

Examples

```
data(aegean2)
```

`coleman` *Fit Coleman's Random Placement Model*

Description

Fit Coleman's (1981) random placement model to a species-site abundance matrix: rows are species and columns are sites. Note that the data must be abundance data and not presence-absence data. According to this model, the number of species occurring on an island depends on the relative area of the island and the regional relative species abundances. The fit of the random placement model can be determined through use of a diagnostic plot (see `plot.coleman`) of island area (log transformed) against species richness, alongside the model's predicted values (see Wang et al., 2010). Following Wang et al. (2010), the model is rejected if more than a third of the observed data points fall beyond one standard deviation from the expected curve.

Usage

```
coleman(data, area)
```

Arguments

`data`	A dataframe or matrix in which rows are species and columns are sites. Each element/value in the matrix is the abundance of a given species in a given site.
`area`	A vector of site (island) area values. The order of the vector must match the order of the columns in `data`.

Value

A list of class "coleman" with four elements. The first element contains the fitted values of the model. The second element contains the standard deviations of the fitted values, and the third and fourth contain the relative island areas and observed richness values, respectively. `plot.coleman` plots the model.

References

Coleman, B. D. (1981). On random placement and species-area relations. Mathematical Biosciences, 54, 191-215.

Matthews, T. J., Cottee-Jones, H. E. W., & Whittaker, R. J. (2015). Quantifying and interpreting nestedness in habitat islands: a synthetic analysis of multiple datasets. Diversity and Distributions, 21, 392-404.

Wang, Y., Bao, Y., Yu, M., Xu, G., & Ding, P. (2010). Nestedness for different reasons: the distributions of birds, lizards and small mammals on islands of an inundated lake. Diversity and Distributions, 16, 862-873.

Examples

```
data(cole_sim)
fit <- coleman(cole_sim[[1]], cole_sim[[2]])
plot(fit, ModTitle = "Hetfield")
```

`cole_sim` *A simulated species-site abundance matrix with site areas*

Description

A dataset in the correct sars format:

Usage

```
data(cole_sim)
```

Format

A list with two elements. The first element contains a species-site abundance matrix in which the rows are species, and the columns are sites/islands. Each value in the matrix is the abundance of a species at a given site. The second element contains a vector of the areas of each site.

Source

Matthews et al. 2015.

Examples

```
data(cole_sim)
```

---

display_sars_models *Display the model information table*

Description

Display Table 1 of Matthews et al. (2019). See sar_multi for further information.

Usage

```
display_sars_models()
```

Value

A table of model information for 21 SAR models, including the model function, number of parameters and general model shape. This includes the 20 models in Matthews et al. (2019); however, note that the mmf model has now been deprecated, and the standard logistic model listed in Tjorve (2003) added instead. Note also, an error in the Chapman Richards model equation has now been corrected, and the shape of some of the models have been updated from sigmoid to convex/sigmoid.

References

Matthews et al. (2019) sars: an R package for fitting, evaluating and comparing species–area relationship models. Ecography, 42, 1446-1455.

Tjørve, E. (2003) Shapes and functions of species–area curves: a review of possible models. Journal of Biogeography, 30, 827-835.

---

galap *A SAR dataset describing the plants of the Galapagos Islands*

Description

A sample dataset in the correct sars format: contains the areas of a number of islands in the Galapagos, and the number of plant species recorded on each island.

Usage

```
data(galap)
```

Format

Adata frame with 2 columns and 16 rows. Each row contains the area of an island (km2) in the Galapagos (1st column) and the number of plants on that island (2nd column).Preston (1962) also includes the island of Albemarle, but we have excluded this as it is almost six times larger than the second largest island.

Source

Preston FW 1962. The Canonical Distribution of Commonness and Rarity: Part I. – Ecology 43:185-215.

Examples

```
data(galap)
```

---

gdm *Fit the General Dynamic Model of Island Biogeography*

Description

Fit the general dynamic model (GDM) of island biogeography using a variety of non-linear and linear SAR models. Functions are provided to compare the GDM fitted using different SAR models, and also, for a given SAR model, to compare the GDM with alternative nested candidate models (e.g. S ~ Area + Time).

Usage

```
gdm(data, model = "linear", mod_sel = FALSE, AST = c(1, 2, 3),
  start_vals = NULL)
```

Arguments

`data`	A dataframe or matrix with at least three columns, where one column should include island area values, one island richness values and one island age values.
`model`	Name of the SAR model to be used to fit the GDM. Can be any of 'loga', 'linear', 'power_area', 'power_area_time', 'all', or 'ATT2'.
`mod_sel`	Logical argument specifying whether, for a given SAR model, a model comparison of the GDM with other nested candidate models should be undertaken.
`AST`	The column locations in `data` for the area, richness and time values (in that order).
`start_vals`	An optional dataframe with starting parameter values for the non-linear regression models (same format as in `nls`). Default is set to NULL.

Details

The GDM models island species richness as a function of island area and island age, and takes the general form: S ~ A + T + T^2, where S = richness, A =area, and T = island age. The T^2 term is included as the GDM predicts a hump-shaped relationship between island richness and island age. However, a variety of different SAR models have been used to fit the GDM and five options are available here: four using non-linear regression and one using linear regression.

Non-linear models

Four SAR models can be used here to fit the GDM: the logarithmic (`model = "loga"`), linear (`model = "linear"`) and power (`model = "power_area"`) SAR models. Another variant of the GDM includes power functions of both area and time (`model = "power_area_time"`). Model fitting follows the procedure in Cardoso et al. (2015). For example, when the linear SAR model is used, the GDM can be fitted using the expression: S ~ Int + A*Area + Ti*T + Ti2*T^2, where Int, A, Ti and Ti2 are free parameters to be estimated. When the power model is used just for area, the equivalent expression is: S ~ exp(Int + A*log(Area) + Ti*T + Ti2*T^2). For all four models, the GDM is fitted using non-linear regression and the `nls` function. It should be noted that the two power models are fitted using S ~ exp(...) to ensure the same response variable (i.e. S and not log(S)) is used in all GDM models and thus AIC etc can be used to compare them.

For each model fit, the residual standard error (RSE), R2 and AIC and AICc values are reported. However, as the model fit object is returned, it is possible to calculate or extract various other measures of goodness of fit (see `nls`).

If `mod_sel = TRUE`, the GDM (using a particular SAR model) is fitted and compared with three other (nested) candidate models: area and time (i.e. no time^2 term), just area, and an intercept only model. The intercept only model is fitted using `lm` rather than `nls`. If `model = "all"`, the GDM is fitted four times (using the power_area, power_area_time, loga and linear SAR models), and the fits compared using `AIC` and `AICc`.

Non-linear regression models are sensitive to the starting parameter values selected. The defaults used here have been chosen as they provide a sensible general choice, but they will not work in all circumstances. As such, alternative starting values can be provided using the `start_vals` argument - this is done in the same way as for `nls`. The four parameter names are: Int (intercept), A (area), Ti (Time), Ti2 (Time^2) (see the example below). This only works for the full GDM non-linear models, and not for the nested models that are fitted when `mod_sel = TRUE` or for the linear models (where they are not needed). If used with `model = "all"`, the same starting parameter values will be provided to each of the four GDM models (power_area, power_area_time, logarithmic and linear).

Linear ATT2 Model

As an alternative to fitting the GDM using non-linear regression, the model can be fitted in various ways using linear regression. This can also be useful if you are having problems with the non-linear regression algorithms not converging. If `model = "ATT2"` is used, the GDM is fitted using the semi-log logarithmic SAR model using linear regression (with untransformed richness and time, and log(area)); this is the original GDM model fitted by Whittaker et al. (2008) and we have used their chosen name (ATT2) to represent it. Steinbauer et al. (2013) fitted variants of this model using linear regression by log-transforming richness and / or time. While we do not provide functionality for fitting these variants, this is easily done by simply providing the log-transformed variable values to the function rather than the untransformed values. Using `model = "ATT2"` is basically a wrapper for the `lm` function. If `mod_sel == TRUE`, the GDM is fitted and compared with three other (nested) candidate models: log(area) and time (i.e. no time^2 term), just log(area), and an intercept only model.

Value

Different objects are returned depending on whether the non-linear or linear regression models are fitted.

Non-linear models

An object of class 'gdm'. If `model` is one of "loga", "linear", "power_area" or "power_area_time" the returned object is a `nls` model fit object. If `model == "all"`, the returned object is a list with four elements; each element being a `nls` fit object. If `mod_sel == TRUE` and `model != "all"`, a list with four elements is returned; each element being a `lm` or `nls` fit object. When `model == "all"`, a list with four elements is returned; each element being a list of the four model fits for a particular SAR model.

Linear ATT2 Model

If `model = "ATT2"` is used, the returned object is of class 'gdm' and 'lm' and all of the method functions associated with standard 'lm' objects (e.g. plot and summary) can be used. If `mod_sel = TRUE` a list with four elements is returned; each element being a `lm` object.

Note

The intercept (Int) parameter that is returned in the power models fits (`model = "power_area" | "power_area_time"`) is on the log scale.

References

Whittaker, R. J., Triantis, K. A., & Ladle, R. J. (2008). A general dynamic theory of oceanic island biogeography. Journal of Biogeography, 35, 977-994.

Borregaard, M. K. et al. (2017). Oceanic island biogeography through the lens of the general dynamic model: assessment and prospect. Biological Reviews, 92, 830-853.

Cardoso, P., Rigal, F., & Carvalho, J. C. (2015). BAT–Biodiversity Assessment Tools, an R package for the measurement and estimation of alpha and beta taxon, phylogenetic and functional diversity. Methods in Ecology and Evolution, 6, 232-236.

Steinbauer, M.J., Dolos, K., Field, R., Reineking, B. & Beierkuhnlein, C. (2013) Re-evaluating the general dynamic theory of oceanic island biogeography. Frontiers of Biogeography, 5.

Carey, M., Boland, J., Weigelt, P. & Keppel, G. (2020) Towards an extended framework for the general dynamic theory of biogeography. Journal of Biogeography, 47, 2554-2566.

Examples

```
#create an example dataset and fit the GDM using the logarithmic SAR model
data(galap)
galap$t <- c(4, 1, 13, 16, 15, 2, 6, 4, 5, 11, 3, 9, 8, 10, 12, 7)
g <- gdm(galap, model = "loga", mod_sel = FALSE)

#Compare the GDM (using the logarithmic model) with other nested candidate
#models
g2 <- gdm(galap, model = "loga", mod_sel = TRUE)

#compare the GDM fitted using the linear, logarithmic and both power models
g3 <- gdm(galap, model = "all", mod_sel = FALSE)
```

```
#fit the GDM using the original ATT2 model of Whittaker et al. 2008 using lm,
#and compare it with other nested models
g4 <- gdm(galap, model = "ATT2", mod_sel = TRUE)

#provide different starting parameter values when fitting the non-linear
#power model GDM
g5 <- gdm(galap, model = "power_area",
start_vals = data.frame("Int" = 0, "A" = 1, Ti = 1, Ti2 = 0))
```

---

get_coef *Calculate the intercepts and slopes of the different segments*

Description

Calculate the intercepts and slopes of the different segments in any of the fitted breakpoint regression models available in the package.

Usage

```
get_coef(fit)
```

Arguments

fit	An object of class 'thresholds', generated using the sar_threshold function.

Details

The coefficients in the fitted breakpoint regression models do not all represent the intercepts and slopes of the different segments; to get these it is necessary to add different coefficients together.

Value

A dataframe with the intercepts (ci) and slopes (zi) of all segments in each fitted model. The numbers attached to c and z relate to the segment, e.g. c1 and z1 are the intercept and slope of the first segment. For the left-horizontal models, the slope of the first segment (i.e. the horizontal segment) is not returned. NA values represent cases where a given parameter is not present in a particular model.

Examples

```
data(aegean2)
a2 <- aegean2[1:168,]
fitT <- sar_threshold(data = a2, mod = c("ContOne", "DiscOne", "ZslopeOne"),
interval = 0.1, non_th_models = TRUE, logAxes = "area", logT = log10)
#get the slopes and intercepts for these three models
coefs <- get_coef(fitT)
coefs
```

lin_pow *Fit the log-log version of the power model*

Description

Fit the log-log version of the power model to SAR data and return parameter values, summary statistics and the fitted values.

Usage

```
lin_pow(data, con = 1, logT = log, compare = FALSE, normaTest =
  "none", homoTest = "none", homoCor = "spearman")
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`con`	The constant to add to the species richness values in cases where one of the islands has zero species.
`logT`	The log-transformation to apply to the area and richness values. Can be any of `log`(default), `log2` or `log10`.
`compare`	Fit the standard (non-linear) power model and return the z-value for comparison (default: `compare = FALSE`).
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of "lillie" (Lilliefors Kolmogorov-Smirnov test), "shapiro" (Shapiro-Wilk test of normality), "kolmo" (Kolmogorov-Smirnov test), or "none" (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of "cor.fitted" (a correlation of the residuals with the model fitted values), "cor.area" (a correlation of the residuals with the area values), or "none" (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest != "none"`. Can be any of "spearman" (the default), "pearson", or "kendall".

Details

A check is made for any islands with zero species. If any zero species islands are found, a constant (default: `con = 1`) is added to each species richness value to enable log transformation. Natural logarithms are used as default, but log2 and log10 can be used instead using the `logT` argument.

The `compare` argument can be used to compare the c and z values calculated using the log-log power model with that calculated using the non-linear power model. Note that the log-log function returns log(c).

Value

A list of class "sars" with up to seven elements. The first element is an object of class 'summary.lm'. This is the summary of the linear model fit using the lm function and the user's data. The second element is a numeric vector of the model's fitted values, and the third contains the log-transformed observed data. The remaining elements depend on the function arguments selected and can include the results of the non-linear power model fit, the log-transformation function used (i.e. `logT`) and the results of any residuals normality and heterogeneity tests.

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model.

Examples

```
data(galap)
fit <- lin_pow(galap, con = 1)
summary(fit)
plot(fit)
```

`niering` *A SAR dataset describing the plants of the Kapingamarangi Atoll*

Description

A sample dataset in the correct sars format: contains the areas of a number of islands in the Kapingamarangi Atoll, and the number of plant species recorded on each island.

Usage

```
data(niering)
```

Format

A data frame with 2 columns and 32 rows. Each row contains the area of an island (km2) in the Kapingamarangi Atoll (1st column) and the number of plants on that island (2nd column).

Source

Niering, W.A. (1963). Terrestrial ecology of Kapingamarangi Atoll, Caroline Islands. Ecol. Monogr., 33, 131–160.

Examples

```
data(niering)
```

plot.coleman *Plot Model Fits for a 'coleman' Object*

Description

S3 method for class 'coleman'. `plot.coleman` creates a plot for objects of class coleman, using the R base plotting framework.

Usage

```
## S3 method for class 'coleman'
plot(
  x,
  xlab = "Relative area (log transformed)",
  ylab = "Species richness",
  pch = 16,
  cex = 1.2,
  pcol = "black",
  cex.lab = 1.3,
  cex.axis = 1,
  lwd = 2,
  lcol1 = "black",
  lcol2 = "darkgrey",
  ModTitle = NULL,
  TiAdj = 0,
  TiLine = 0.5,
  cex.main = 1.5,
  ...
)
```

Arguments

`x`	An object of class 'coleman'.
`xlab`	Title for the x-axis.
`ylab`	Title for the y-axis.
`pch`	Plotting character (for points).
`cex`	A numerical vector giving the amount by which plotting symbols (points) should be scaled relative to the default.
`pcol`	Colour of the points.
`cex.lab`	The amount by which the the axis titles should be scaled relative to the default.
`cex.axis`	The amount by which the the axis labels should be scaled relative to the default.
`lwd`	Line width.
`lcol1`	Line colour of the fitted model curve.
`lcol2`	Line colour of the model standard deviation curves.

ModTitle	Plot title (default is null, which equates to no main title).
TiAdj	Which way the plot title (if included) is justified.
TiLine	Places the plot title (if included) this many lines outwards from the plot edge.
cex.main	The amount by which the the plot title (if included) should be scaled relative to the default.
...	Further graphical parameters (see par, plot.default,title, lines) may be supplied as arguments.

Details

The resultant plot contains the observed richness values with the model fit and confidence intervals. Following Wang et al. (2010), the model is rejected if more than a third of the observed data points fall beyond one standard deviation from the expected curve.

Examples

```
data(cole_sim)
fit <- coleman(cole_sim[[1]], cole_sim[[2]])
plot(fit, ModTitle = "Hetfield")
```

---

plot.multi *Plot Model Fits for a 'multi' Object*

Description

S3 method for class 'multi'. `plot.multi` creates plots for objects of class multi, using the R base plotting framework. Plots of all model fits, the multimodel SAR curve (with confidence intervals) and a barplot of the information criterion weights of the different models can be constructed.

Usage

```
## S3 method for class 'multi'
plot(
  x,
  type = "multi",
  allCurves = TRUE,
  xlab = NULL,
  ylab = NULL,
  pch = 16,
  cex = 1.2,
  pcol = "dodgerblue2",
  ModTitle = NULL,
  TiAdj = 0,
  TiLine = 0.5,
  cex.main = 1.5,
  cex.lab = 1.3,
```

```
  cex.axis = 1,
  yRange = NULL,
  lwd = 2,
  lcol = "dodgerblue2",
  mmSep = FALSE,
  lwd.Sep = 6,
  col.Sep = "black",
  pLeg = TRUE,
  modNames = NULL,
  cex.names = 0.88,
  subset_weights = NULL,
  confInt = FALSE,
  ...
)
```

Arguments

x	An object of class 'multi'.
type	The type of plot to be constructed: either `type = multi` for a plot of the multimodel SAR curve, or `type = bar` for a barplot of the information criterion weights of each model.
allCurves	A logical argument for use with `type = multi` that specifies whether all the model fits should be plotted with the multimodel SAR curve (`allCurves = TRUE`; the default) or that only the multimodel SAR curve should be plotted (`allCurves = FALSE`).
xlab	Title for the x-axis. Only for use with `type = multi`.
ylab	Title for the y-axis.
pch	Plotting character (for points). Only for use with `type = multi`.
cex	A numerical vector giving the amount by which plotting symbols (points) should be scaled relative to the default.
pcol	Colour of the points. Only for use with `type = multi`.
ModTitle	Plot title (default is `ModTitle = NULL`, which reverts to "Multimodel SAR" for `type = multi` and to "Model weights" for `type = bar`). For no title, use `ModTitle = ""`.
TiAdj	Which way the plot title is justified.
TiLine	Places the plot title this many lines outwards from the plot edge.
cex.main	The amount by which the plot title should be scaled relative to the default.
cex.lab	The amount by which the axis titles should be scaled relative to the default.
cex.axis	The amount by which the axis labels should be scaled relative to the default.
yRange	The range of the y-axis. Only for use with `type = multi`.
lwd	Line width. Only for use with `type = multi`.
lcol	Line colour. Only for use with `type = multi`.

mmSep	Logical argument of whether the multimodel curve should be plotted as a separate line (default = FALSE) on top of the others, giving the user more control over line width and colour. Only for use with `type = multi` and `allCurves = TRUE`.
lwd.Sep	If `mmSep = TRUE`, the line width of the multimodel curve.
col.Sep	If `mmSep = TRUE`, the colour of the multimodel curve.
pLeg	Logical argument specifying whether or not the legend should be plotted (when `type = multi` and `allCurves = TRUE`).
modNames	A vector of model names for the barplot of weights (when `type = bar`). The default (`modNames = NULL`) uses abbreviated versions (see below) of the names from the `sar_average` function.
cex.names	The amount by which the axis labels (model names) should be scaled relative to the default. Only for use with `type = bar`.
subset_weights	Only create a barplot of the model weights for models with a weight value above a given threshold (`subset_weights`). Only for use with `type = bar`.
confInt	A logical argument specifying whether confidence intervals should be plotted around the multimodel curve. Can only be used if confidence intervals have been generated in the `sar_average` function.
...	Further graphical parameters (see `par`, `plot.default`,`title`, `lines`) may be supplied as arguments.

Note

In some versions of R and R studio, when plotting all model fits on the same plot with a legend it is necessary to manually extend your plotting window (height and width; e.g. the 'Plots' window of R studio) before plotting to ensure the legend fits in the plot. Extending the plotting window after plotting sometimes just stretches the legend.

Occasionally a model fit will converge and pass the model fitting checks (e.g. residual normality) but the resulting fit is nonsensical (e.g. a horizontal line with intercept at zero). Thus, it can be useful to plot the resultant 'multi' object to check the individual model fits. To re-run the `sar_average` function without a particular model, simply remove it from the `obj` argument.

For visual interpretation of the model weights barplot it is necessary to abbreviate the model names when plotting the weights of several models. To plot fewer bars, use the `subset_weights` argument to filter out models with lower weights than a threshold value. To provide a different set of names use the `modNames` argument. The model abbreviations used as the default are:

- Pow = Power
- PowR = PowerR
- E1 = Extended_Power_model_1
- E2 = Extended_Power_model_2
- P1 = Persistence_function_1
- P2 = Persistence_function_2
- Loga = Logarithmic
- Kob = Kobayashi

- MMF = MMF
- Mon = Monod
- NegE = Negative_exponential
- CR = Chapman_Richards
- CW3 = Cumulative_Weibull_3_par.
- AR = Asymptotic_regression
- RF = Rational_function
- Gom = Gompertz
- CW4 = Cumulative_Weibull_4_par.
- BP = Beta-P_cumulative
- Logi = Logistic(Standard)
- Hel = Heleg(Logistic)
- Lin = Linear_model

Examples

```
data(galap)
#plot a multimodel SAR curve with all model fits included
fit <- sar_average(data = galap, grid_start = "none")
plot(fit)

#remove the legend
plot(fit, pLeg = FALSE)

#plot just the multimodel curve
plot(fit, allCurves = FALSE, ModTitle = "", lcol = "black")

#plot all model fits and the multimodel curve on top as a thicker line
plot(fit, allCurves = TRUE, mmSep = TRUE, lwd.Sep = 6, col.Sep = "orange")

#Plot a barplot of the model weights
plot(fit, type = "bar")
#subset to plot only models with weight > 0.05
plot(fit, type = "bar", subset_weights = 0.05)
```

`plot.sars` *Plot Model Fits for a 'sars' Object*

Description

S3 method for class 'sars'. `plot.sars` creates plots for objects of class 'sars' (type = 'fit', "lin_pow' and 'fit_collection'), using the R base plotting framework. The exact plot(s) constructed depends on the 'Type' attribute of the 'sars' object. For example, for a 'sars' object of Type 'fit', the `plot.sars` function returns a plot of the model fit (line) and the observed richness values (points). For a 'sars' object of Type 'fit_collection' the `plot.sars` function returns either a grid with n individual plots (corresponding to the n model fits in the fit_collection), or a single plot with all n model fits included.

For plotting a 'sar_average' object, see `plot.multi`.

Usage

```
## S3 method for class 'sars'
plot(
  x,
  mfplot = FALSE,
  xlab = NULL,
  ylab = NULL,
  pch = 16,
  cex = 1.2,
  pcol = "dodgerblue2",
  ModTitle = NULL,
  TiAdj = 0,
  TiLine = 0.5,
  cex.main = 1.5,
  cex.lab = 1.3,
  cex.axis = 1,
  yRange = NULL,
  lwd = 2,
  lcol = "dodgerblue2",
  di = NULL,
  pLeg = FALSE,
  ...
)
```

Arguments

`x`	An object of class 'sars'.
`mfplot`	Logical argument specifying whether the model fits in a fit_collection should be plotted on one single plot (`mfplot = TRUE`) or separate plots (`mfplot = FALSE`; the default).
`xlab`	Title for the x-axis (default depends on the Type attribute).
`ylab`	Title for the y-axis (default depends on the Type attribute).
`pch`	Plotting character (for points).
`cex`	A numerical vector giving the amount by which plotting symbols (points) should be scaled relative to the default.
`pcol`	Colour of the points.
`ModTitle`	Plot title (default is `ModTitle = NULL`, which reverts to a default name depending on the type of plot). For no title, use `ModTitle = ""`. For a sars object of type fit_collection, a vector of names can be provided (e.g. `letters[1:3]`).
`TiAdj`	Which way the plot title is justified.
`TiLine`	Places the plot title this many lines outwards from the plot edge.

`cex.main`	The amount by which the plot title should be scaled relative to the default.
`cex.lab`	The amount by which the axis titles should be scaled relative to the default.
`cex.axis`	The amount by which the axis labels should be scaled relative to the default.
`yRange`	The range of the y-axis.
`lwd`	Line width.
`lcol`	Line colour.
`di`	Dimensions to be passed to `par(mfrow=())` to specify the size of the plotting window, when plotting multiple plots from a sars object of Type fit_collection. For example, `di = c(1, 3)` creates a plotting window with 1 row and 3 columns. The default (null) creates a square plotting window of the correct size.
`pLeg`	Logical argument specifying whether or not the legend should be plotted for fit_collection plots (when `mfplot = TRUE`) or. When a large number of model fits are plotted the legend takes up a lot of space, and thus the default is `pLeg = FALSE`.
`...`	Further graphical parameters (see `par`, `plot.default`,`title`, `lines`) may be supplied as arguments.

Examples

```
data(galap)
#fit and plot a sars object of Type fit.
fit <- sar_power(galap)
plot(fit, ModTitle = "A)", lcol = "blue")

#fit and plot a sars object of Type fit_collection.
fc <- sar_multi(data = galap, obj = c("power", "loga", "epm1"),
grid_start = "none")
plot(fc, ModTitle = letters[1:3], xlab = "Size of island")
```

`plot.threshold` *Plot Model Fits for a 'threshold' Object*

Description

S3 method for class 'threshold'. `plot.threshold` creates plots for objects of class threshold, using the R base plotting framework. Plots of single or multiple threshold models can be constructed.

Usage

```
## S3 method for class 'threshold'
plot(
  x,
  xlab = NULL,
  ylab = NULL,
  multPlot = TRUE,
```

```
    pch = 16,
    cex = 1.2,
    pcol = "black",
    ModTitle = NULL,
    TiAdj = 0,
    TiLine = 0.5,
    cex.main = 1.5,
    cex.lab = 1.3,
    cex.axis = 1,
    yRange = NULL,
    lwd = 2,
    lcol = "red",
    di = NULL,
    ...
)
```

Arguments

`x`	An object of class 'threshold'.
`xlab`	Title for the x-axis. Defaults will depend on any axes log-transformations.
`ylab`	Title for the y-axis.Defaults will depend on any axes log-transformations.
`multPlot`	Whether separate plots should be built for each model fit (default = TRUE) or all model fits should be printed on the same plot (FALSE)
`pch`	Plotting character (for points).
`cex`	A numerical vector giving the amount by which plotting symbols (points) should be scaled relative to the default.
`pcol`	Colour of the points.
`ModTitle`	Plot title (default is `ModTitle = NULL`), which reverts to the model names. For no title, use `ModTitle = ""`.
`TiAdj`	Which way the plot title is justified.
`TiLine`	Places the plot title this many lines outwards from the plot edge.
`cex.main`	The amount by which the plot title should be scaled relative to the default.
`cex.lab`	The amount by which the axis titles should be scaled relative to the default.
`cex.axis`	The amount by which the axis labels should be scaled relative to the default.
`yRange`	The range of the y-axis. Default taken as the largest value bacross the observed and fitted values.
`lwd`	Line width.
`lcol`	Line colour. If `multPlot = TRUE`, just a single colour should be given, If `multPlot = FALSE`, either a single colour, or a vector of colours the same length as the number of model fits in `x`.
`di`	Dimensions to be passed to `par(mfrow=())` to specify the size of the plotting window, when plotting multiple plots. For example, `di = c(1, 3)` creates a plotting window with 1 row and 3 columns. The default (`NULL`) creates a plotting window large enough to fit all plots in.
`...`	Further graphical parameters (see `par`, `plot.default`,`title`, `lines`) may be supplied as arguments.

Note

The raw lm model fit objects are returned with the sar_threshold function if the user wishes to construct their own plots.

Use par(mai = c()) prior to calling plot, to set the graph margins, which can be useful when plotting multiple models in a single plot to ensure space within the plot taken up by the individual model fit plots is maximised.

Examples

```
data(aegean)

#fit two threshold models (in logA-S space) and the linear and
#intercept only models
fct <- sar_threshold(aegean, mod = c("ContOne", "DiscOne"),
                    non_th_models = TRUE,  interval = 5,
                    parallel = FALSE, logAxes = "area")

#plot using default settings
plot(fct)

#change various plotting settings, and set the graph margins prior to
#plotting
par(mai = c(0.7,0.7, 0.4, 0.3))
plot(fct, pcol = "blue", pch = 18, lcol = "green",
    ModTitle = c("A", "B", "C", "D"), TiAdj = 0.5, xlab = "Yorke")

#Plot multiple model fits in the same plot, with different colour for each
#model fit
plot(fct, multPlot = FALSE, lcol = c("black", "red", "green", "purple"))
```

---

sars_models *Display the 21 SAR model names*

Description

Display the 21 SAR model names as a vector. See sar_multi for further information.

Usage

```
sars_models()
```

Value

A vector of model names.

Note

sar_mmf is included here for now but has been deprecated (see News) sar_asymp *Fit the Asymptotic regression model*

Description

Fit the Asymptotic regression model to SAR data.

Usage

```
sar_asymp(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by

assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The summary.sars function returns a more useful summary of the model fit results, and the plot.sars plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_asymp(galap)
summary(fit)
plot(fit)
```

---

sar_average   *Fit a multimodel averaged SAR curve*

Description

Construct a multimodel averaged species-area relationship curve using information criterion weights and up to twenty SAR models.

Usage

```
sar_average(obj = c("power",
  "powerR","epm1","epm2","p1","p2","loga","koba",
  "monod","negexpo","chapman","weibull3","asymp",
  "ratio","gompertz","weibull4","betap","logistic", "heleg", "linear"), data =
  NULL, crit = "Info", normaTest = "none", homoTest = "none", homoCor =
  "spearman", neg_check = FALSE, alpha_normtest = 0.05, alpha_homotest =
  0.05, grid_start = "partial", grid_n = NULL, confInt = FALSE, ciN = 100,
  verb = TRUE, display = TRUE)
```

Arguments

obj	Either a vector of model names or a fit_collection object created using `sar_multi`. If a vector of names is provided, `sar_average` first calls `sar_multi` before generating the averaged multimodel curve.
data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site. If `obj` is a fit_collection object, `data` should be NULL.
crit	The criterion used to compare models and compute the model weights. The default `crit = "Info"` switches to AIC or AICc depending on the number of data points in the dataset. AIC (`crit = "AIC"`) or AICc (`crit = "AICc"`) can be chosen regardless of the sample size. For BIC, use `crit ="Bayes"`.
normaTest	The test used to test the normality of the residuals of each model. Can be any of "lillie" (Lilliefors Kolmogorov-Smirnov test), "shapiro" (Shapiro-Wilk test of normality), "kolmo" (Kolmogorov-Smirnov test), or "none" (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of each model. Can be any of "cor.fitted" (a correlation of the squared residuals with the model fitted values), "cor.area" (a correlation of the squared residuals with the area values), or "none" (no residuals homogeneity test is undertaken; the default).

homoCor The correlation test to be used when `homoTest != "none"`. Can be any of "spearman" (the default), "pearson", or "kendall".

neg_check Whether or not a check should be undertaken to flag any models that predict negative richness values.

alpha_normtest The alpha value used in the residual normality test (default = 0.05, i.e. any test with a P value < 0.05 is flagged as failing the test).

alpha_homotest The alpha value used in the residual homogeneity test (default = 0.05, i.e. any test with a P value < 0.05 is flagged as failing the test).

grid_start Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.

grid_n If `grid_start = exhaustive`, the number of points sampled in the starting parameter space (see details).

confInt A logical argument specifying whether confidence intervals should be calculated for the multimodel curve using bootstrapping.

ciN The number of bootstrap samples to be drawn to calculate the confidence intervals (if `confInt == TRUE`).

verb verbose - Whether or not to print certain warnings (default: `verb == TRUE`)

display Show the model fitting output and related messages. (default: `display == TRUE`).

Details

The multimodel SAR curve is constructed using information criterion weights (see Burnham & Anderson, 2002; Guilhaumon et al. 2010). If `obj` is a vector of `n` model names the function fits the `n` models to the dataset provided using the `sar_multi` function. A dataset must have four or more datapoints to fit the multimodel curve. If any models cannot be fitted they are removed from the multimodel SAR. If `obj` is a fit_collection object (created using the `sar_multi` function), any model fits in the collection which are NA are removed. In addition, if any other model checks have been selected (i.e. residual normality and heterogeneity tests, and checks for negative predicted richness values), these are undertaken and any model that fails the selected test(s) is removed from the multimodel SAR. The order of the additional checks inside the function is (if all are turned on): normality of residuals, homogeneity of residuals, and a check for negative fitted values. Once a model fails one test it is removed and thus is not available for further tests. Thus, a model may fail multiple tests but the returned warning will only provide information on a single test. We have now changed the defaults so that no checks are undertaken, so it is up to the user to select any checks if appropriate.

The resultant models are then used to construct the multimodel SAR curve. For each model in turn, the model fitted values are multiplied by the information criterion weight of that model, and the resultant values are summed across all models (Burnham & Anderson, 2002). Confidence intervals can be calculated (using `confInt`) around the multimodel averaged curve using the bootstrap procedure outlined in Guilhaumon et al (2010).The procedure transforms the residuals from the individual model fits and occasionally NAs / Inf values can be produced - in these cases, the model is removed from the confidence interval calculation (but not the multimodel curve itself). There is also a constraint within the procedure to remove any transformed residuals that result in negative richness values. When several SAR models are used, when grid_start is turned on and when the number of bootstraps (ciN) is large, generating the confidence intervals can take a (very) long time. Parallel processing will be added to future versions.

Choosing starting parameter values for non-linear regression optimisation algorithms is not always straight forward, depending on the data at hand. In the package, we use various approaches to choose default starting parameters. However, we also use a grid search process which creates a large array of different possible starting parameter values (within certain bounds) and then randomly selects a proportion of these to test. There are three options for the grid_start argument to control this process. The default (grid_start = "partial") randomly samples 500 different sets of starting parameter values for each model, adds these to the model's default starting values and tests all of these. A more comprehensive set of starting parameter estimates can be used (grid_start = "exhaustive") - this option allows the user to choose the number of starting parameter sets to be tested (using the grid_n argument) and includes a range of additional starting parameter estimates, e.g. very small values and particular values we have found to be useful for individual models. Using grid_start = "exhaustive" in combination with a large grid_n can be very time consuming; however, we would recommend it as it makes it more likely that the optimal model fit will be found, particularly for the more complex models. This is particularly true if any of the model fits does not converge, returns a singular gradient at parameter estimates, or the plot of the model fit does not look optimum. The grid start procedure can also be turned off (grid_start = "none"), meaning just the default starting parameter estimates are used. Note that grid_start has been disabled for a small number of models (e.g. Weibull 3 par.). See the vignette for more information. Remember that, as grid_start has a random component, when grid_start != "none", you can get slightly different results each time you fit a model or run sar_average.

Even with grid_start, occasionally a model fit will be able to be fitted and pass the model fitting checks (e.g. residual normality) but the resulting fit is nonsensical (e.g. a horizontal line with intercept at zero). Thus, it can be useful to plot the resultant 'multi' object to check the individual model fits. To re-run the sar_average function without a particular model, simply remove it from the obj argument.

The sar_models() function can be used to bring up a list of the 20 model names. display_sars_models() generates a table of the 20 models with model information.

Value

A list of class "multi" and class "sars" with two elements. The first element ('mmi') contains the fitted values of the multimodel sar curve. The second element ('details') is a list with the following components:

- mod_names Names of the models that were successfully fitted and passed any model check
- fits A fit_collection object containing the successful model fits
- ic The information criterion selected
- norm_test The residual normality test selected
- homo_test The residual homogeneity test selected
- alpha_norm_test The alpha value used in the residual normality test
- alpha_homo_test The alpha value used in the residual homogeneity test
- ics The information criterion values (e.g. AIC values) of the model fits
- delta_ics The delta information criterion values

- weights_ics The information criterion weights of each model fit
- n_points Number of data points
- n_mods The number of successfully fitted models
- no_fit Names of the models which could not be fitted or did not pass model checks
- convergence Logical value indicating whether `optim` model convergence code = 0, for each model

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.multi` plots the multimodel curve.

Note

There are different types of non-convergence and these are dealt with differently in the package. If the optimisation algorithm fails to return any solution, the model fit is defined as NA and is then removed, and so does not appear in the model summary table or multi-model curve etc. However, the optimisation algorithm (e.g. Nelder-Mead) can also return non-NA model fits but where the solution is potentially non-optimal (e.g. degeneracy of the Nelder–Mead simplex) - these cases are identified by any `optim` convergence code that is not zero. We have decided not to remove these fits (i.e. they are kept in the model summary table and multimodel curve) - as arguably a non-optimal fit is still better than no fit - but any instances can be checked using the returned `details$converged` vector and then the model fitting re-run without these models, if preferred. Increasing the starting parameters grid search (see above) may also help avoid this issue.

The generation of confidence intervals around the multimodel curve (using `confInt == TRUE`), may throw up errors that we have yet to come across. Please report any issues to the package maintainer.

There are different formulas for calculating the various information criteria (IC) used for model comparison (e.g. AIC, BIC). For example, some formulas use the residual sum of squares (rss) and others the log-likelihood (ll). Both are valid approaches and will give the same parameter estimates, but it is important to only compare IC values that have been calculated using the same approach. For example, the 'sars' package used to use formulas based on the rss, while the nls function function in the stats package uses formulas based on the ll. To increase the compatibility between nls and sars, we have changed our formulas such that now our IC formulas are the same as those used in the nls function. See the "On the calculation of information criteria" section in the package vignette for more information.

The mmf model was found to be equivalent to the He & Legendre logistic, and so the former has been deprecated (as of Feb 2021). We have removed it from the default models in `sar_average`, although it is still available to be used for the time being (using the `obj` argument). The standard logistic model has been added in its place, and is now used as default within `sar_average`.

References

Burnham, K. P., & Anderson, D. R. (2002). Model selection and multi-model inference: a practical information-theoretic approach (2nd ed.). New-York: Springer.

Guilhaumon, F., Mouillot, D., & Gimenez, O. (2010). mmSAR: an R-package for multimodel species-area relationship inference. Ecography, 33, 420-424.

Matthews, T. J., K. A. Triantis, R. J. Whittaker, & F. Guilhaumon. (2019). sars: an R package for fitting, evaluating and comparing species–area relationship models. Ecography, 42, 1446–55.

Examples

```
data(galap)
#attempt to construct a multimodel SAR curve using all twenty sar models
#using no grid_start just for speed here (not recommended generally)
fit <- sar_average(data = galap, grid_start = "none")
summary(fit)
plot(fit)

# construct a multimodel SAR curve using a fit_collection object
ff <- sar_multi(galap, obj = c("power", "loga", "monod", "weibull3"))
fit2 <- sar_average(obj = ff, data = NULL)
summary(fit2)

## Not run:
# construct a multimodel SAR curve using a more exhaustive set of starting
# parameter values
fit3 <- sar_average(data = galap, grid_start = "exhaustive", grid_n = 1000)

## End(Not run)
```

---

sar_betap *Fit the Beta-P cumulative model*

Description

Fit the Beta-P cumulative model to SAR data.

Usage

```
sar_betap(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).

homoTest The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).

homoCor The correlation test to be used when homoTest !='none'. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.

verb Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the optim function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the start argument), or a more comprehensive search can be undertaken using the grid_start argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model

- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
#Grid_start turned off for speed (not recommended)
data(galap)
fit <- sar_betap(galap, grid_start = 'none')
summary(fit)
plot(fit)
```

`sar_chapman` *Fit the Chapman Richards model*

Description

Fit the Chapman Richards model to SAR data.

Usage

```
sar_chapman(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares

- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_chapman(galap)
summary(fit)
plot(fit)
```

sar_epm1 *Fit the Extended Power model 1 model*

Description

Fit the Extended Power model 1 model to SAR data.

Usage

```
sar_epm1(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by

assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The summary.sars function returns a more useful summary of the model fit results, and the plot.sars plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_epm1(galap)
summary(fit)
plot(fit)
```

---

sar_epm2 *Fit the Extended Power model 2 model*

Description

Fit the Extended Power model 2 model to SAR data.

Usage

```
sar_epm2(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test

- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_epm2(galap)
summary(fit)
plot(fit)
```

`sar_gompertz` *Fit the Gompertz model*

Description

Fit the Gompertz model to SAR data.

Usage

```
sar_gompertz(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).

homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model

- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_gompertz(galap)
summary(fit)
plot(fit)
```

`sar_heleg` *Fit the Heleg(Logistic) model*

Description

Fit the Heleg(Logistic) model to SAR data.

Usage

```
sar_heleg(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares

- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_heleg(galap)
summary(fit)
plot(fit)
```

`sar_koba` *Fit the Kobayashi model*

Description

Fit the Kobayashi model to SAR data.

Usage

```
sar_koba(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by

assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The summary.sars function returns a more useful summary of the model fit results, and the plot.sars plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_koba(galap)
summary(fit)
plot(fit)
```

`sar_linear` *Fit the linear model*

Description

Fit the linear model to SAR data.

Usage

```
sar_linear(data, normaTest =  'none', homoTest = 'none', homoCor =
  'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors Kolmogorov-Smirnov test), 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest != "none"`. Can be any of "spearman" (the default), "pearson", or "kendall".
`verb`	Whether or not to print certain warnings (default = TRUE).

Details

The model is fitted using linear regression and the `lm` function. Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails.

A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- verge Logical code indicating model convergence
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

Examples

```
data(galap)
fit <- sar_linear(galap)
summary(fit)
plot(fit)
```

`sar_loga` *Fit the Logarithmic model*

Description

Fit the Logarithmic model to SAR data.

Usage

```
sar_loga(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares

- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_loga(galap)
summary(fit)
plot(fit)
```

`sar_logistic` *Fit the Logistic(Standard) model*

Description

Fit the Logistic(Standard) model to SAR data.

Usage

```
sar_logistic(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by

assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The summary.sars function returns a more useful summary of the model fit results, and the plot.sars plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_logistic(galap)
summary(fit)
plot(fit)
```

`sar_mmf` *Fit the MMF model*

Description

Fit the MMF model to SAR data. This function has been deprecated.

Usage

```
sar_mmf(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test

- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- suppressWarnings(sar_mmf(galap))
summary(fit)
plot(fit)
```

`sar_monod` *Fit the Monod model*

Description

Fit the Monod model to SAR data.

Usage

```
sar_monod(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).

homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model

- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_monod(galap)
summary(fit)
plot(fit)
```

`sar_multi` *Create a Collection of SAR Model Fits*

Description

Creates a fit collection of SAR model fits, which can then be plotted using `plot.sars`.

Usage

```
sar_multi(data, obj = c("power",
  "powerR","epm1","epm2","p1","p2","loga","koba",
  "monod","negexpo","chapman","weibull3","asymp",
  "ratio","gompertz","weibull4","betap","logistic","heleg","linear"),
  normaTest = "none", homoTest = "none", homoCor = "spearman", grid_start =
  "partial", grid_n = NULL, verb = TRUE, display = TRUE)
```

Arguments

- `data` — A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
- `obj` — A vector of model names.
- `normaTest` — The test used to test the normality of the residuals of each model. Can be any of "lillie" (Lilliefors Kolmogorov-Smirnov test), "shapiro" (Shapiro-Wilk test of normality), "kolmo" (Kolmogorov-Smirnov test), or "none" (no residuals normality test is undertaken; the default).
- `homoTest` — The test used to check for homogeneity of the residuals of each model. Can be any of "cor.fitted" (a correlation of the squared residuals with the model fitted values), "cor.area" (a correlation of the squared residuals with the area values), or "none" (no residuals homogeneity test is undertaken; the default).
- `homoCor` — The correlation test to be used when `homoTest != "none"`. Can be any of "spearman" (the default), "pearson", or "kendall".
- `grid_start` — Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
- `grid_n` — If `grid_start = exhaustive`, the number of points sampled in the starting parameter space (see details).
- `verb` — verbose - Whether or not to print certain warnings (default: `verb == TRUE`).
- `display` — Show the model fitting output and related messages. (default: `display == TRUE`).

Details

The `sar_models()` function can be used to bring up a list of the 20 model names. `display_sars_models()` generates a table of the 20 models with model information.

Value

A list of class 'sars' with n elements, corresponding to the n individual SAR model fits.

Examples

```
data(galap)
# construct a fit_collection object of 3 SAR model fits
fit2 <- sar_multi(galap, obj = c("power", "loga", "linear"))
plot(fit2)

# construct a fit_collection object of all 20 SAR model fits
# using no grid_start for speed
fit3 <- sar_multi(galap, grid_start = "none")
```

sar_negexpo *Fit the Negative exponential model*

Description

Fit the Negative exponential model to SAR data.

Usage

```
sar_negexpo(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by

assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The summary.sars function returns a more useful summary of the model fit results, and the plot.sars plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_negexpo(galap)
summary(fit)
plot(fit)
```

---

sar_p1 *Fit the Persistence function 1 model*

Description

Fit the Persistence function 1 model to SAR data.

Usage

```
sar_p1(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test

- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_p1(galap)
summary(fit)
plot(fit)
```

`sar_p2` *Fit the Persistence function 2 model*

Description

Fit the Persistence function 2 model to SAR data.

Usage

```
sar_p2(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).

homoTest The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).

homoCor The correlation test to be used when homoTest !='none'. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.

verb Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the optim function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the start argument), or a more comprehensive search can be undertaken using the grid_start argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model

- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_p2(galap)
summary(fit)
plot(fit)
```

`sar_power` *Fit the Power model*

Description

Fit the Power model to SAR data.

Usage

```
sar_power(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If grid_start = exhaustive, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when homoTest !='none'. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the optim function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the start argument), or a more comprehensive search can be undertaken using the grid_start argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error. For the power model (and only this model) the returned object (sigConf) and model summary also includes the parameter estimates generated from fitting the model using nls and using as starting parameter estimates the parameter values from our model fitting. This also returns the confidence intervals generated with confint (which calls MASS:::confint.nls), which should be more accurate than the default sars CIs.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_power(galap)
summary(fit)
plot(fit)
```

sar_powerR *Fit the PowerR model*

Description

Fit the PowerR model to SAR data.

Usage

```
sar_powerR(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If `grid_start` = `exhaustive`, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by

assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The summary.sars function returns a more useful summary of the model fit results, and the plot.sars plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_powerR(galap)
summary(fit)
plot(fit)
```

---

sar_pred — *Use SAR model fits to predict richness on islands of a given size*

Description

Predict the richness on an island of a given size using either individual SAR model fits, a fit_collection of model fits, or a multi-model SAR curve.

Usage

```
sar_pred(fit, area)
```

Arguments

fit	Either a model fit object, a fit_collection object (generated using sar_multi), or a sar_multi object (generated using sar_average).
area	A numeric vector of area values (length >= 1).

Details

Extrapolation (e.g. predicting the richness of areas too large to be sampled) is one of the primary uses of the SAR. The `sar_pred` function provides an easy method for undertaking such an exercise. The function works by taking an already fitted SAR model, extacting the parameter values and then using these values and the model function to predict the richness for any value of area provided.

If a multi-model SAR curve is used for prediction (i.e. using sar_average), the model information criterion weight (i.e. the conditional probabilities for each of the n models) for each of the individual model fits that were used to generate the curve are stored. The n models are then each used to predict the richness of a larger area and these predictions are multiplied by the respective model weights and summed to provide a multi-model averaged prediction.

Value

A data.frame of class 'sars' with three columns: 1) the name of the model, 2) the area value for which a prediction has been generated, and 3) the prediction from the model extrapolation.

Note

This function is used in the ISAR extrapolation paper of Matthews & Aspin (2019).

Code to calculate confidence intervals around the predictions using bootstrapping will be added in a later version of the package.

As grid_start has a random component, when `grid_start != "none"` in your model fitting, you can get slightly different results each time you fit a model or run `sar_average` and then run `sar_pred` on it. We would recommend using `grid_start = "exhaustive"` as this is more likely to find the optimum fit for a given model.

References

Matthews, T.J. & Aspin, T.W.H. (2019) Model averaging fails to improve the extrapolation capability of the island species–area relationship. Journal of Biogeography, 46, 1558-1568.

Examples

```
data(galap)
#fit the power model and predict richness on an island of area = 5000
fit <- sar_power(data = galap)
p <- sar_pred(fit, area = 5000)

#fit three SAR models and predict richness on islands of area = 5000 & 10000
#using no grid_start for speed
fit2 <- sar_multi(galap, obj = c("power", "loga", "koba"), grid_start = "none")
p2 <- sar_pred(fit2, area = c(5000, 10000))

#calculate a multi-model curve and predict richness on islands of area = 5000 & 10000
#using no grid_start for speed
fit3 <- sar_average(data = galap, grid_start = "none")
p3 <- sar_pred(fit3, area = c(5000, 10000))
```

`sar_ratio` *Fit the Rational function model*

Description

Fit the Rational function model to SAR data.

Usage

```
sar_ratio(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

data	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
start	NULL or custom parameter start values for the optimisation algorithm.
grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares

- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_ratio(galap)
summary(fit)
plot(fit)
```

`sar_threshold` *Fit threshold SAR models*

Description

Fit up to six piecewise (threshold) regression models to SAR data.

Usage

```
sar_threshold(data, mod = "All", interval = NULL, nisl = NULL,
  non_th_models = TRUE, logAxes = "area", con = 1, logT = log, parallel =
  FALSE, cores = NULL)
```

Arguments

`data`	A dataset in the form of a dataframe with at least two columns: the first with island/site areas, and the second with the species richness of each island/site.
`mod`	A vector of model names: an individual model, a set of models, or all models. Can be any of 'All' (fit all models), 'ContOne' (continuous one-threshold), 'ZslopeOne' (left-horizontal one-threshold), 'DiscOne' (discontinuous one-threshold), 'ContTwo' (continuous two-threshold), 'ZslopeTwo' (left-horizontal two-threshold), or 'DiscTwo' (discontinuous two-threshold).
`interval`	The amount to increment the threshold value by in the iterative model fitting process (not applicable for the discontinuous models). The default for non-transformed area reverts to 1, while for log-transformed area it is 0.01. However, these values may not be suitable depending on the range of area values in a dataset, and thus users are advised to manually set this argument.
`nisl`	Set the minimum number of islands to be contained within each of the two segments (in the case of one-threshold models), or the first and last segments (in the case of two-threshold models). It needs to be less than than half of the total number of islands in the dataset. Default = NULL.
`non_th_models`	Logical argument (default = TRUE) of whether two non-threshold models (i.e. a simple linear regression: y ~ x; and an intercept only model: y ~ 1) should also be fitted.
`logAxes`	What log-transformation (if any) should be applied to the area and richness values. Should be one of "none" (no transformation), "area" (only area is log-transformed; default) or "both" (both area and richness log-transformed).
`con`	The constant to add to the species richness values in cases where one of the islands has zero species.
`logT`	The log-transformation to apply to the area and richness values. Can be any of `log`(default), `log2` or `log10`.
`parallel`	Logical argument for whether parallel processing should be used. Only applicable when the continuous two-threshold and left-horizontal two-threshold models are being fitted.
`cores`	Number of cores to use. Only applicable when `parallel = TRUE`.

Details

This function is described in more detail in the accompanying paper (Matthews & Rigal, 2020).

Fitting the continuous and left-horizontal piecewise models (particularly the two-threshold models) can be time consuming if the range in area is large and/or the `interval` argument is small. For the two-threshold continuous slope and left-horizontal models, the use of parallel processing (using the `parallel` argument) is recommended. The number of cores (`cores`) must be provided.

Note that the interval argument is not used to fit discontinuous models, as, in these cases, the breakpoint must be at a datapoint.

There has been considerable debate regarding the number of parameters that are included in different piecewise models. Here (and thus in our calculation of AIC, AICc, BIC etc) we consider ContOne to have five parameters, ZslopeOne - 4, DiscOne - 6, ContTwo - 7, ZslopeTwo - 6, DiscTwo - 8. The standard linear model and the intercept model are considered to have 3 and 2 parameters, respectively. The raw `lm` model fits are provided in the output, however, if users want to calculate information criteria using different numbers of parameters.

The raw `lm` model fits can also be used to explore classic diagnostic plots for linear regression analysis in R using the function `plot` or other diagnostic tests such `outlierTest`, `leveragePlots` or `influencePlot`, available in the `car` package. This is advised as currently there are no model validation checks undertaken automatically, unlike elsewhere in the sars package.

Confidence intervals around the breakpoints in the one-threshold continuous and left- horizontal models can be calculated using the `threshold_ci` function. The intercepts and slopes of the different segments in the fitted breakpoint models can be calculated using the `get_coef` function.

Rarely, multiple breakpoint values can return the same minimum rss (for a given model fit). In these cases, we just randomly choose and return one and also produce a warning. If this occurs it is worth checking the data and model fits carefully.

The `nisl` argument can be useful to avoid situations where a segment contains only one island, for example. However, setting strict criteria on the number of data points to be included in segments could be seen as "forcing" the fit of the model, and arguably if a model fit is not interpretable, it is simply that the model does not provide a good representation of the data. Thus, it should not be used without careful thought.

Value

A list of class "threshold" and "sars" with five elements. The first element contains the different model fits (lm objects). The second element contains the names of the fitted models, the third contains the threshold values, the fourth element the dataset (i.e. a dataframe with area and richness values), and the fifth contains details of any axes log-transformations undertaken. `summary.sars` provides a more user-friendly ouput (including a model summary table) and `plot.threshold` plots the model fits.

Note

Due to the increased number of parameters, fitting piecewise regression models to datasets with few islands is not recommended. In particular, we would advise against fitting the two-threshold models to small SAR datasets (e.g. fewer than 10 islands for the one threshold models, and 20 islands for the two threshold models).

Author(s)

Francois Rigal and Thomas J. Matthews

References

Lomolino, M.V. & Weiser, M.D. (2001) Towards a more general species-area relationship: diversity on all islands, great and small. Journal of Biogeography, 28, 431-445.

Gao, D., Cao, Z., Xu, P. & Perry, G. (2019) On piecewise models and species-area patterns. Ecology and Evolution, 9, 8351-8361.

Matthews, T.J. et al. (2020) Unravelling the small-island effect through phylogenetic community ecology. Journal of Biogeography.

Matthews, T.J. & Rigal, F. (In Review) Thresholds and the species–area relationship: a set of functions for fitting, evaluating and plotting a range of commonly used piecewise models. Frontiers of Biogeography.

Examples

```
data(aegean2)
a2 <- aegean2[1:168,]
fitT <- sar_threshold(data = a2, mod = c("ContOne", "DiscOne"),
interval = 0.1, non_th_models = TRUE, logAxes = "area", logT = log10)
summary(fitT)
plot(fitT)
#diagnostic plots for the ContOne model
par(mfrow=c(2, 2))
plot(fitT[[1]][[1]])
```

`sar_weibull3` *Fit the Cumulative Weibull 3 par. model*

Description

Fit the Cumulative Weibull 3 par. model to SAR data.

Usage

```
sar_weibull3(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.

grid_start	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
grid_n	If grid_start = exhaustive, the number of points sampled in the starting parameter space.
normaTest	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
homoTest	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
homoCor	The correlation test to be used when homoTest !='none'. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
verb	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the optim function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the start argument), or a more comprehensive search can be undertaken using the grid_start argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (normaTest) and homogeneity (homoTest) of the residuals and a warning is provided in summary.sars if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also sar_average).

As grid_start has a random component, when grid_start != 'none' in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares
- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found

- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_weibull3(galap)
summary(fit)
plot(fit)
```

---

sar_weibull4 *Fit the Cumulative Weibull 4 par. model*

Description

Fit the Cumulative Weibull 4 par. model to SAR data.

Usage

```
sar_weibull4(data, start = NULL, grid_start = 'partial',
  grid_n = NULL, normaTest = 'none',
  homoTest = 'none', homoCor = 'spearman', verb = TRUE)
```

Arguments

`data`	A dataset in the form of a dataframe with two columns: the first with island/site areas, and the second with the species richness of each island/site.
`start`	NULL or custom parameter start values for the optimisation algorithm.
`grid_start`	Should a grid search procedure be implemented to test multiple starting parameter values. Can be one of 'none', 'partial' or 'exhaustive' The default is set to 'partial'.
`grid_n`	If `grid_start = exhaustive`, the number of points sampled in the starting parameter space.
`normaTest`	The test used to test the normality of the residuals of the model. Can be any of 'lillie' (Lilliefors test , 'shapiro' (Shapiro-Wilk test of normality), 'kolmo' (Kolmogorov-Smirnov test), or 'none' (no residuals normality test is undertaken; the default).
`homoTest`	The test used to check for homogeneity of the residuals of the model. Can be any of 'cor.fitted' (a correlation of the residuals with the model fitted values), 'cor.area' (a correlation of the residuals with the area values), or 'none' (no residuals homogeneity test is undertaken; the default).
`homoCor`	The correlation test to be used when `homoTest !='none'`. Can be any of 'spearman' (the default), 'pearson', or 'kendall'.
`verb`	Whether or not to print certain warnings (default = TRUE)

Details

The model is fitted using non-linear regression. The model parameters are estimated by minimizing the residual sum of squares with an unconstrained Nelder-Mead optimization algorithm and the `optim` function. To avoid numerical problems and speed up the convergence process, the starting values used to run the optimization algorithm are carefully chosen. However, if this does not work, custom values can be provided (using the `start` argument), or a more comprehensive search can be undertaken using the `grid_start` argument. See the vignette for more information. The fitting process also determines the observed shape of the model fit, and whether or not the observed fit is asymptotic (see Triantis et al. 2012 for further details). Model validation can be undertaken by assessing the normality (`normaTest`) and homogeneity (`homoTest`) of the residuals and a warning is provided in `summary.sars` if either test is chosen and fails. A selection of information criteria (e.g. AIC, BIC) are returned and can be used to compare models (see also `sar_average`).

As grid_start has a random component, when `grid_start != 'none'` in your model fitting, you can get slightly different results each time you fit a model

The parameter confidence intervals returned in sigConf are just simple confidence intervals, calculated as 2 * standard error.

Value

A list of class 'sars' with the following components:

- par The model parameters
- value Residual sum of squares

- counts The number of iterations for the convergence of the fitting algorithm
- convergence Numeric code returned from optim indicating model convergence (0 = converged)
- message Any message from the model fit algorithm
- hessian A symmetric matrix giving an estimate of the Hessian at the solution found
- verge Logical code indicating that optim model convergence value is zero
- startValues The start values for the model parameters used in the optimisation
- data Observed data
- model A list of model information (e.g. the model name and formula)
- calculated The fitted values of the model
- residuals The model residuals
- AIC The AIC value of the model
- AICc The AICc value of the model
- BIC The BIC value of the model
- R2 The R2 value of the model
- R2a The adjusted R2 value of the model
- sigConf The model coefficients table
- normaTest The results of the residuals normality test
- homoTest The results of the residuals homogeneity test
- observed_shape The observed shape of the model fit
- asymptote A logical value indicating whether the observed fit is asymptotic
- neg_check A logical value indicating whether negative fitted values have been returned

The `summary.sars` function returns a more useful summary of the model fit results, and the `plot.sars` plots the model fit.

References

Triantis, K.A., Guilhaumon, F. & Whittaker, R.J. (2012) The island species-area relationship: biology and statistics. Journal of Biogeography, 39, 215-231.

Examples

```
data(galap)
fit <- sar_weibull4(galap)
summary(fit)
plot(fit)
```

summary.sars *Summarising the results of the model fitting functions*

Description

S3 method for class 'sars'. summary.sars creates summary statistics for objects of class 'sars'. The exact summary statistics computed depends on the 'Type' attribute (e.g. 'multi') of the 'sars' object. The summary method generates more useful information for the user than the standard model fitting functions. Another S3 method (print.summary.sars; not documented) is used to print the output.

Usage

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

Arguments

object	An object of class 'sars'.
...	Further arguments.

Value

The summary.sars function returns an object of class "summary.sars". A print function is used to obtain and print a summary of the model fit results.

For a 'sars' object of Type 'fit', a list with 16 elements is returned that contains useful information from the model fit, including the model parameter table (with t-values, p-values and confidence intervals), model fit statistics (e.g. R2, AIC), the observed shape of the model and whether or not the fit is asymptotic, and the results of any additional model checks undertaken (e.g. normality of the residuals).

For a 'sars' object of Type 'multi', a list with 5 elements is returned: (i) a vector of the names of the models that were successfully fitted and passed any additional checks, (ii) a character string containing the name of the criterion used to rank models, (iii) a data frame of the ranked models, (iv) a vector of the names of any models that were not fitted or did not pass any additional checks, and (v) a logical vector specifying whether the optim convergence code for each model that passed all the checks is zero. In regards to (iii; Model_table), the dataframe contains the fit summaries for each successfully fitted model (including the value of the model criterion used to compare models, the R2 and adjusted R2, and the observed shape of the fit); the models are ranked in decreasing order of information criterion weight.

For a 'sars' object of Type 'lin_pow', a list with up to 7 elements is returned: (i) the model fit output from the lm function, (ii) the fitted values of the model, (iii) the observed data, (iv and v) the results of the residuals normality and heterogeneity tests, and (vi) the log-transformation function used. If the argument compare = TRUE is used in lin_pow, a 7th element is returned that contains the parameter values from the non-linear power model.

For a 'sars' object of Type 'threshold', a list with three elements is returned: (i) the information criterion used to order the ranked model summary table (currently just BIC), (ii) a model summary table (models are ranked using BIC), and (iii) details of any axes log-transformations undertaken.

Note that in the model summary table, if log-area is used as the predictor, the threshold values will be on the log scale used. Thus it may be preferable to back-transform them (e.g. using `exp(th)` if natural logarithms are used) so that they are on the scale of untransformed area. Th1 and Th2 in the table are the threshold value(s), and seg1, seg2, seg3 provide the number of datapoints within each segment (for the threshold models); one-threshold models have two segements, and two-threshold models have three segments.

Examples

```
data(galap)
#fit a multimodel SAR and get the model table
mf <- sar_average(data = galap, grid_start = "none")
summary(mf)
summary(mf)$Model_table
#Get a summary of the fit of the linear power model
fit <- lin_pow(galap, con = 1, compare = TRUE)
summary(fit)
```

---

threshold_ci *Calculate confidence intervals around breakpoints*

Description

Generate confidence intervals around the breakpoints of the one-threshold continuous and left-horizontal models. Two types of confidence interval can be implemented: a confidence interval derived from an inverted F test and an empirical bootstrap confidence interval.

Usage

```
threshold_ci(object, cl = 0.95, method = "boot", interval = NULL,
  Nboot = 100, verb = TRUE)
```

Arguments

object	An object of class ’thresholds’, generated using the sar_threshold function. The object must contain fits of either (or both) of the one-threshold continuous or the one-threshold left-horizontal model.
cl	The confidence level. Default value is 0.95 (95 percent).
method	Either bootstraping (boot) or inverted F test (F).
interval	The amount to increment the threshold value by in the iterative model fitting process used in both the F and boot methods. The default for non-transformed area reverts to 1, while for log-transformed area it is 0.01. It is advised that the same interval value used when running sar_threshold is used here.
Nboot	Number of bootstrap samples (for use with method = "boot").
verb	Should progress be reported. If TRUE, every 50th bootstrap sample is reported (for use with method = "boot").

Details

Full details of the two approaches can be found in Toms and Lesperance (2003). If the number of bootstrap samples is large, the function can take a while to run. Following Toms and Lesperance (2003), we therefore recommend the use of the inverted F test confidence interval when sample size is large, and bootstrapped confidence intervals when sample size is smaller.

Currently only available for the one-threshold continuous and left- horizontal threshold models.

Value

A list of class "sars" with two elements. If method “F” is used, the list contains only the confidence interval values. If method “boot” is used, the list contains two elements. The first element is the full set of bootstrapped breakpoint estimates for each model and the second contains the confidence interval values.

Author(s)

Francois Rigal and Christian Paroissin

References

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

Examples

```
data(aegean2)
a2 <- aegean2[1:168,]
fitT <- sar_threshold(data = a2, mod = "ContOne",
interval = 0.1, non_th_models = TRUE, logAxes = "area", logT = log10)
#calculate confidence intervals using bootstrapping
#(very low Nboot just as an example)
CI <- threshold_ci(fitT, method = "boot", interval = NULL, Nboot = 3)
CI
#Use the F method instead, with 90% confidence interval
CI2 <- threshold_ci(fitT, cl = 0.90, method = "F", interval = NULL)
CI2
```

Index