Package ‘dendrometry’

October 8, 2025

Type Package

Date 2025-10-06

Title Forest Estimations and Dendrometric Computations

Version 0.0.4

Description Computation of dendrometric and structural parameters from forest inventory data. The objective is to provide a user-friendly R package for researchers, ecologists, foresters, statisticians, loggers and other persons who deal with forest inventory data. The package includes advanced distribution fitting capabilities with multiple estimation methods (Maximum Likelihood, Maximum Product Spacing with ties correction methods following Cheng & Amin (1983), and Method of Moments) for probability distributions commonly used in forestry. Visualization tools with confidence bands using delta method and parametric bootstrap are provided for three-parameter Weibull distribution fitting to diameter data. Useful conversion of angle value from degree to radian, conversion from angle to slope (in percentage) and their reciprocals as well as principal angle determination are also included. Position and dispersion parameters usually found in forest studies are implemented. The package contains Fibonacci series, its extensions and the Golden Number computation. Useful references are Arcadius Y. J. Akossou, Soufianou Arzouma, Eloi Y. Attakpa, Noël H. Fonton and Kouami Kokou (2013) <doi:10.3390/d5010099>, W. Bonou, R. Glele Kakaï, A.E. Assogbadjo, H.N. Fonton, B. Sinsin (2009) <doi:10.1016/j.foreco.2009.05.032>, R. C. H. Cheng and N. A. K. Amin (1983) <doi:1 6161.1983.tb01268.x>, and R. C. H. Cheng and M. A. Stephens (1989) <doi:10.1093/biomet/76.2.385>.

Depends R (>= 3.5.0)

VignetteBuilder knitr

Suggests knitr, rmarkdown

Imports graphics, grDevices

Encoding UTF-8

BugReports `https://github.com/narcisstar/dendrometry-issues/issues`

LazyData true

RoxygenNote 7.3.3

Config/usethis/last-upkeep 2025-04-23

NeedsCompilation no

Author Narcisse Yehouenou [aut, cre],
Information and Communication Technology for you ONG (ICT4U-ONG) [fnd]

Maintainer Narcisse Yehouenou <`narcisstar211@gmail.com`>

Repository CRAN

Date/Publication 2025-10-08 14:50:02 UTC

Contents

`AIC.fit_dist` *Extract AIC*

Description

S3 method to extract the Akaike Information Criterion from a fitted distribution.

Usage

```
## S3 method for class 'fit_dist'
AIC(object, ..., k = 2)
```

Arguments

`object`	an object of class "fit_dist".
`...`	additional arguments (not used).
`k`	penalty per parameter (default: 2 for AIC).

Value

AIC value.

`angle2slope` *Angle - Slope conversion and Principal Measure determination*

Description

Conversion of angle to slope values and reciprocally.
`angle2slope` converts angle to slope values.
`slope2angle` converts slope to angle values.
`principal` determines the principal measure of an angle value. Principal measure ranges from -pi to pi for radian unit while it ranges from -180 to 180 for degree unit.

Usage

```
angle2slope(angle, angleUnit = c("deg", "rad"))

slope2angle(slope, angleUnit = c("deg", "rad"))

principal(angle, angleUnit = c("deg", "rad"))
```

Arguments

`angle`	numeric, vector of angle to be converted to slope.
`angleUnit`	character, unit of `angle`. For `slope2angle`, the desired unit for the returned angle value. For `principal`, both the angle input and output unit. Either `deg` or `rad`. Default is `deg`.
`slope`	numeric, vector of slope to be converted to angle.

Value

Object of class `angle`.
`angle2slope` returns vector of slope values while `slope2angle` and `principal` return vector of angle values in unit specified in `angle` argument.

Note

Use `principal` in position computations, not distance computations.

See Also

`deg` and `rad`.

Examples

```
angle2slope(10)
angle2slope(angle = 45)
angle2slope(angle = pi / 4, angleUnit = "rad")
angle2slope(1.047198, "rad")
angle2slope(seq(0.2, 1.5, .4), angleUnit = "rad") #'

slope2angle(100)
slope2angle(100, "rad")
round(pi / 4, 2)

slope2angle(17.6327)
slope2angle(angle2slope(30))

principal(303)
principal(23 * pi / 8, "rad")
principal(7 * pi / 4, angleUnit = "rad")
deg(principal(7 * pi / 4, angleUnit = "rad"))
principal(7 * 45)
```

`barkFactor` *Bark factor The bark factor (k) is computed for trees in order to assess the importance of the valuable wood in the overall volume of a tree (Husch et al., 1982):*

Description

Bark factor The bark factor (k) is computed for trees in order to assess the importance of the valuable wood in the overall volume of a tree (Husch et al., 1982):

Usage

```
barkFactor(dbh, thickness)
```

Arguments

`dbh`	numeric, diameter over bark of the individual trees.
`thickness`	numeric, bark thickness measured on individual trees.

References

Husch, B., Miller, C., Beers, T., 1982. Forest mensuration. Ronald Press Company, London, pp. 1 – 410.

`basal` *The basal area of plots*

Description

Computes the basal area of tree stems in a plot. The basal area is the cross sectional area of the bole or stem of a tree at breast height.

Usage

```
basal(dbh, area, k = 100, circum = NULL)
```

Arguments

`dbh`	numeric, vector of diameter.
`area`	numeric, area of the plot (see `details` for unit).
`k`	numeric, used to convert diameter unit. Default is `100` (coverts from cm to m. See `details`).
`circum`	numeric, vector of circumference. Is used only if `dbh` is not given.

Details

If `area` is expressed in ha and `dbh` expressed in cm, the basal area unit is cm\u00b2/ha when `k = 1`. In order to convert centimeter (cm) to meter (m) for `dbh`, set `k = 100`. Because 1m = 100 cm. Then, basal area unit will be `m\u00b2/ha`.

If `dbh` is in meter (m), and `area` in in hectare (ha), setting `k = 1` returns basal area in m\u00b2/ha.

If `dbh` is in feet, and `area` in acre, setting `k = 1` returns basal area in ft\u00b2/ac.

If `dbh` is in inch, and `area` in acre, setting `k = 12` returns basal area in feet\u00b2/acres (ft\u00b2/ac).

Value

A vector of basal area of stands.

`basalContribution` *Basal area contribution*

Description

The basal area contribution (in per cent) is defined as the part of a given species trees in the overall basal area of all trees in an area.

Usage

```
basalContribution(basal)
```

Arguments

`basal`	numeric, basal area per species.

`basal_i` *Individual Basal Area and DBH (diameter)*

Description

`basal_i` computes the basal area of a tree stem (individual), the area of a circle of diameter `dbh`. `basal2dbh` computes the dbh (diameter) based on the basal area.

Usage

```
basal_i(dbh, circum = NULL)

basal2dbh(basal)
```

Arguments

`dbh`	numeric, vector of diameter.
`circum`	numeric, vector of circumference. Is used only if `dbh` is not given.
`basal`	numeric, individual basal area.

Details

If `circum` is given, `dbh` is not used.

Value

`basal_i` returns individual basal area while `basal2dbh` returns DBH.

Examples

```
basal_i(dbh = 10)
basal_i(circum = 31.41)
basal2dbh(78.53982)
```

`BIC.fit_dist` *Extract BIC*

Description

S3 method to extract the Bayesian Information Criterion from a fitted distribution.

Usage

```
## S3 method for class 'fit_dist'
BIC(object, ...)
```

Arguments

`object`	an object of class "fit_dist".
`...`	additional arguments (not used).

Value

BIC value.

`blackman` *Index of Blackman*

Description

Index of Blackman

Usage

```
blackman(density)
```

Arguments

`density`	numeric, vector of the density.

Value

Index of Blackman.

`coef.fit_dist` *Extract Model Coefficients*

Description

S3 method to extract parameter estimates from a fitted distribution.

Usage

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

Arguments

`object`	an object of class "fit_dist".
`...`	additional arguments (not used).

Value

Named numeric vector of parameter estimates.

`confint.fit_dist` *Confidence Intervals for Parameters*

Description

S3 method to compute confidence intervals for distribution parameters based on asymptotic normality of maximum likelihood estimates.

Usage

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

Arguments

`object`	an object of class "fit_dist".
`parm`	character vector of parameter names or numeric vector of indices. If missing, all parameters are considered.
`level`	confidence level (default: 0.95).
`...`	additional arguments (not used).

Value

A matrix with columns giving lower and upper confidence limits for each parameter.

`dbh` *Diameter (DBH) and Circumference*

Description

`DBH` computes diameter (at breast height) based on circumference (at breast height).
`circum` computes circumference (at breast height) based on diameter (at breast height).
They are based on circle diameter and perimeter formulas.

Usage

```
dbh(circum)

circum(dbh)
```

Arguments

`circum`	numeric, vector of circumference.
`dbh`	numeric, vector of diameter.

Value

`dbh`, returns diameter and `circum`, returns circumference.

See Also

See also `height` for tree height.

Examples

```
perimeter <- seq(30, 60, 1.4)
diameter <- dbh(perimeter)
circum(diameter)
```

`decrease` *The decrease coefficient*

Description

This coefficient expresses the ratio between the diameter (or circumference) at mid-height of the bole and the diameter (or circumference) measured at breast height.

Usage

```
decrease(middle, breast)
```

Arguments

`middle`	numeric, the diameter or circumference at middle height.
`breast`	numeric, the diameter or circumference at breast height.

Details

Both `middle` and `breast` arguments should be of the same type (either diameter or circumference). Not mixture.

Value

A vector of decrease coefficients.

Examples

```
decrease(30, 120)
decrease(middle = 40, breast = 90)
```

`decreaseMetric` *Metric scrolling or decay*

Description

The average metric decay expresses the difference, in centimeters per meter, between the diameter (or circumference) at breast height and its diameter at mid-height of a stem related to the difference between the height at mid-height and that at breast height.

Usage

```
decreaseMetric(dmh, dbh, mh, bh = 1.3)
```

Arguments

`dmh`	numeric, the diameter at middle height in centimeter (cm).
`dbh`	numeric, the diameter at breast height in centimeter (cm).
`mh`	numeric, the middle (or cut) height in meter (m).
`bh`	Either a numeric value standing for the breast height in meter (m) of all trees or a numeric vector standing for the breast height of each tree. Default is `1.3`.

Value

Metric decay

See Also

`reducecoef`

Examples

```
decreaseMetric(dmh = 40, dbh = 90, mh = 7)
decreaseMetric(45, 85, 9)
```

`deg` *Degree and Radian*

Description

`deg` converts angle values from radians to degrees.
`rad` converts angle values from degrees to radians.

Usage

```
deg(radian)

rad(degree)
```

Arguments

`radian`	numeric, vector of radian values to be converted to degrees.
`degree`	numeric, vector of degree values to be converted to radians.

Value

`deg` returns vector of degree values while `rad` returns vector of radian values.

See Also

`principal`.

Examples

```
deg(pi / 2)
rad(180)
```

`densityRegen` *Density of regeneration (efficient version)*

Description

Computes the density per plot of tree regeneration based on counts in subplots. Can be grouped by additional factors for nested analysis.

Usage

```
densityRegen(data = NULL, plot = NULL, count, nbSubPlot, area, ...)
```

Arguments

`data`	an optional data frame, list, tibble or object coercible by `as.data.frame` to a data frame containing the variables whose names are given in `count` and `plot`.
`plot`	an optional character, name of the variable containing the plot identities. If `data` is missing, a vector providing the plot identities.
`count`	character, name of the variable containing the counts: number of stems (individuals). If `data` is missing, a numeric vector providing the the counts: number of stems (individuals).
`nbSubPlot`	numeric, number of subplots per plot.
`area`	numeric, area of each subplot.
`...`	additional factor variables for grouping (e.g., species, site, treatment)

`densityTree` *Tree density*

Description

Density of trees per plot.

Usage

```
densityTree(number, area, overall = TRUE)
```

Arguments

`number`	numeric, vector of tree count in each plot.
`area`	numeric, area of a plot.
`overall`	logical, if `TRUE`, an overall mean density is computed, otherwise density is computed for each plot. Default is `TRUE`.

Details

If every plot have same area, `area` is a numeric value, otherwise `area` is a vector of each plot area.

Value

Vector of density.

See Also

`densityRegen` for regeneration density.

Examples

```
count <- setNames(
  c(87, 104, 83, 132, 107, 84, 110, 115, 112, 94),
  LETTERS[1:10]
)
densityTree(count, 10)
densityTree(count, area = 10, overall = FALSE)
densityTree(count, area = 10:19, overall = FALSE)
```

`diameterMean` *Mean diameter*

Description

Mean diameter of a forestry stand.

Usage

```
diameterMean(dbh)
```

Arguments

`dbh`	numeric, vector of diameter.

Value

Mean diameter.

See Also

`dbh`, `basal_i`

Examples

```
set.seed(1)
diameter <- rnorm(10, 100, 20)
diameterMean(dbh = diameter)
```

`distanceH` *Horizontal distance*

Description

Horizontal distance calculation for sloping area.

Usage

```
distanceH(
  distance,
  angle,
  type = c("slope", "angle"),
  angleUnit = c("deg", "rad")
)
```

Arguments

`distance`	numeric, vector of the distance measured on sloping area.
`angle`	numeric, vector of angle or slope values.
`type`	character, type of `angle` argument. Either `"angle"` or `"slope"`. Default is `"slope"`.
`angleUnit`	character, unit of `angle` measures if `type = "angle"`. Either `"deg"` for degree or `"rad"` for radian. Default is `"deg"`.

Value

A vector of horizontal distance.

Examples

```
distanceH(20, 30)
distanceH(20, angle = 30, type = "slope")
distanceH(20, angle = 25, type = "angle")
```

`dweibull3` *The three-parameter Weibull Distribution*

Description

Density, distribution function, quantile function and random generation for the three-parameter Weibull.

Usage

```
dweibull3(x, shape, scale = 1, loc = 0, log = FALSE)
pweibull3(q, shape, scale, loc = 0, lower.tail = TRUE, log.p = FALSE)
qweibull3(p, shape, scale, loc = 0, lower.tail = TRUE, log.p = FALSE)
rweibull3(n, shape, scale = 1, loc = 0)
```

Arguments

`x, q`	vector of quantiles.
`shape, scale, loc`	shape, scale and location parameters. The two latter default to `1` and `0` respectively.
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as `log(p)`.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

See Also

`dweibull` for the Weibull distribution.

`factorize` *Making factor vectors*

Description

Changes character vectors of a data set to factor vectors.

Usage

```
factorize(data, binary = FALSE)
```

Arguments

`data`	data frame or tibble data set.
`binary`	logical indicating if binary numeric data should be considered as factor. Default is `FALSE`.

Details

When `binary = TRUE`, variables stored as numeric and which have exactly two levels are changed to factor.

Value

Data frame with all character vectors changed to factor vectors.

`fibonacci` *Fibonacci series*

Description

Generates numbers from Fibonacci series.

Usage

```
fibonacci(n, PrintFib = FALSE, Uo = 0, U1 = 1)
```

Arguments

`n`	integer, the size of the series.
`PrintFib`	logical, indicating if the series should be printed.
`Uo, U1`	integer, the first two numbers of the series.

Details

The series equation is Un = U_(n-2) /U_(n-1).

Value

Either an integer, result of the function or a vector of `n` first numbers of the series.

Author(s)

Narcisse Yehouenou <`narcisstar211@gmail.com`>

See Also

`fiboRate`

Examples

```
fibonacci(n = 10, PrintFib = TRUE)
fibonacci(n = 10, Uo = 1, U1 = 3, PrintFib = FALSE)
```

`fiboRate` *Fibonacci series ratio*

Description

Computes rates from Fibonacci series.

Usage

```
fiboRate(n, PrintSer = FALSE, Uo = 0, U1 = 1)
```

Arguments

`n`	integer, the size of the series.
`PrintSer`	logical, indicating if the series should be printed.
`Uo, U1`	integer, the first number of the series.

Details

The series equation is Un = U_(n-2) /U_(n-1). The function returns golden number when Uo = 0, and U1 = 1. Larger n is, more precise the number (result) is.

Value

Either a numeric, result of the rate of `nth` and `(n-1)th` numbers in Fibonacci series or all `(n-1)th` those rates.

Author(s)

Narcisse Yehouenou <narcisstar211@gmail.com>

See Also

`fibonacci`

Examples

```
## Golden number (Le Nombre d'Or)
fiboRate(n = 18, PrintSer = FALSE, Uo = 0, U1 = 1)
## (1+sqrt(5))/2
fiboRate(n = 10, PrintSer = TRUE, Uo = 0, U1 = 1)
```

`fit_dist` *Unified Parameter Estimation for Probability Distributions*

Description

Estimates parameters of probability distributions using various methods: Maximum Likelihood (MLE), Maximum Product Spacing (MPS), or Method of Moments (MOM).

Usage

```
fit_dist(
  data,
  dist = "normal",
  method = "mle",
  start = NULL,
  lower = NULL,
  upper = NULL,
  optim_method = "Nelder-Mead",
  custom_functions = NULL,
  tol_spacing = 1e-16,
  tol_param = 1e-06,
  ties_method = "cheng_amin",
  ...
)
```

Arguments

`data`	numeric vector of observed data.
`dist`	character string specifying the distribution. Options include: "normal", "exponential", "gamma", "weibull", "weibull3", "lognormal", "lognormal3", "beta", or "custom".

`method`	character string specifying estimation method. Options: "mle" (Maximum Likelihood - default), "mps" (Maximum Product Spacing), "mom" (Method of Moments).
`start`	named list or numeric vector of initial parameter values. Required for "custom" distributions with "mps" or "mle" methods.
`lower`	named list or numeric vector of lower bounds for parameters.
`upper`	named list or numeric vector of upper bounds for parameters.
`optim_method`	optimization method passed to `optim`. Default is "Nelder-Mead".
`custom_functions`	list containing custom distribution functions (for dist="custom"): • `pdf`: probability density function f(x, params) (for MLE and ties correction) • `cdf`: cumulative distribution function F(x, params) (for MPS) • `param_names`: character vector of parameter names • `start_fn`: function to compute starting values (optional)
`tol_spacing`	numeric tolerance for spacings/densities to avoid log(0). Default is 1e-16.
`tol_param`	numeric tolerance for parameter lower bounds. Default is 1e-6.
`ties_method`	character string for handling ties in MPS. Options: "cheng_amin" (default - Cheng & Amin 1983), "none", "cheng_stephens" (Cheng & Stephens 1989). Only applicable when method = "mps".
`...`	Additional arguments passed to `optim`.

Value

A list with class "fit_dist" containing:

`estimate`	Named vector of parameter estimates
`vcov`	Variance-covariance matrix (for mps/mle)
`se`	Standard errors (for mps/mle)
`loglik`	Log-likelihood value
`aic`	Akaike Information Criterion
`bic`	Bayesian Information Criterion
`objective`	Maximum value of objective function (logspacing for mps, loglik for mle)
`ks_statistic`	Kolmogorov-Smirnov test statistic
`ks_pvalue`	Kolmogorov-Smirnov test p-value
`convergence`	Convergence code from optim (0 indicates success)
`message`	Convergence message from optim
`data`	Original data (sorted)
`dist`	Distribution name
`method`	Estimation method used
`n`	Sample size
`k`	Number of parameters
`tol_spacing`	Tolerance used for spacings/densities
`tol_param`	Tolerance used for parameter bounds
`ties_method`	Ties correction method (for MPS only)

References

Cheng, R. C. H., & Amin, N. A. K. (1983). Estimating parameters in continuous univariate distributions with a shifted origin. Journal of the Royal Statistical Society: Series B, 45(3), 394-403.

Cheng, R. C. H., & Stephens, M. A. (1989). A goodness-of-fit test using Moran's statistic with estimated parameters. Biometrika, 76(2), 385-392.

Examples

```
# MLE estimation (default)
set.seed(123)
x <- rweibull(100, shape = 2.5, scale = 1.5)
fit1 <- fit_dist(x, dist = "weibull")
print(fit1)

# MPS estimation with Cheng-Amin ties correction
fit2 <- fit_dist(x, dist = "weibull", method = "mps")

# L-Moments estimation (under development)
# fit3 <- fit_dist(x, dist = "weibull", method = "lm")

# Method of Moments
fit4 <- fit_dist(x, dist = "weibull", method = "mom")

# Compare fits
cat("AIC - MLE:", fit1$aic, "MPS:", fit2$aic, "\n")
```

`fit_weibull_plot` *Fit and Plot Three-Parameter Weibull Distribution with Confidence Bands*

Description

Fits a three-parameter Weibull distribution to diameter data and optionally visualizes the fit with a histogram, fitted density curve, and confidence bands.

Usage

```
fit_weibull_plot(
  x,
  amplitude = 10,
  shape = 2,
  plot = TRUE,
  show_ci = TRUE,
  ci_level = 0.95,
  ci_method = "delta",
  n_boot = 1000,
  main = NULL,
```

```
    title.col = "black",
    mid = TRUE,
    line.col = "blue",
    ci.col = "lightblue",
    ci.alpha = 0.3,
    legendPos = "topright",
    lowLim = NULL,
    ymax = NULL,
    bg = "aliceblue",
    method = "mle",
    ties_method = "cheng_amin",
    cex.axis = 0.6,
    cex.lab = 0.8,
    las = 1,
    xlab = "Diameter class (cm)",
    ylab = "Relative frequency (%)",
    cex.legend = 1,
    ...
  )

  adjWeibull(
    x,
    amplitude = 10,
    shape = 2,
    plot = TRUE,
    show_ci = TRUE,
    ci_level = 0.95,
    ci_method = "delta",
    n_boot = 1000,
    main = NULL,
    title.col = "black",
    mid = TRUE,
    line.col = "blue",
    ci.col = "lightblue",
    ci.alpha = 0.3,
    legendPos = "topright",
    lowLim = NULL,
    ymax = NULL,
    bg = "aliceblue",
    method = "mle",
    ties_method = "cheng_amin",
    cex.axis = 0.6,
    cex.lab = 0.8,
    las = 1,
    xlab = "Diameter class (cm)",
    ylab = "Relative frequency (%)",
    cex.legend = 1,
    ...
```

```
)
```

Arguments

`x`	numeric vector of diameter observations (typically tree diameters in cm).
`amplitude`	numeric bin width for histogram (default: 10).
`shape`	numeric initial value for shape parameter (default: 2).
`plot`	logical; if TRUE, produces a histogram with fitted curve (default: TRUE).
`show_ci`	logical; if TRUE, displays confidence bands around fitted curve (default: TRUE). Only applicable when plot = TRUE and standard errors are available.
`ci_level`	numeric confidence level for bands (default: 0.95).
`ci_method`	character string for CI computation: "delta" (delta method using vcov), "bootstrap" (parametric bootstrap), or "both" (default: "delta").
`n_boot`	integer number of bootstrap samples (default: 1000). Only used if ci_method is "bootstrap" or "both".
`main`	character string for plot title (default: NULL).
`title.col`	color for legend title (default: "black").
`mid`	logical; if TRUE, curve is drawn from min to max of bin midpoints; if FALSE, from min to max of bin breaks (default: TRUE).
`line.col`	color for fitted curve (default: "blue").
`ci.col`	color for confidence bands (default: "lightblue").
`ci.alpha`	numeric transparency for confidence bands (default: 0.3).
`legendPos`	position of legend (default: "topright").
`lowLim`	numeric lower limit for histogram breaks (default: NULL, uses min(x)).
`ymax`	numeric upper limit for y-axis (default: NULL, auto-computed).
`bg`	background color for legend box (default: "aliceblue").
`method`	character string specifying estimation method. Options: "mle" (Maximum Likelihood - default), "mps" (Maximum Product Spacing), "mom" (Method of Moments).
`ties_method`	character string for ties correction in MPS: "cheng_amin", "none", or "cheng_stephens" (default: "cheng_amin"). Only used when method = "mps".
`cex.axis`	numeric character expansion factor for axis annotation (default: 0.6).
`cex.lab`	numeric character expansion factor for axis labels (default: 0.8).
`las`	numeric orientation of axis labels (default: 1).
`xlab`	character string for x-axis label (default: "Diameter class (cm)").
`ylab`	character string for y-axis label (default: "Relative frequency (%)").
`cex.legend`	numeric character expansion factor for legend (default: 1).
`...`	additional graphical parameters passed to `plot`.

Value

A list (invisibly) containing:

`estimate`	Named vector of parameter estimates (shape, scale, location)
`se`	Standard errors (NULL if unavailable or method doesn't support it)
`vcov`	Variance-covariance matrix (NULL if unavailable)
`ci`	Confidence intervals for parameters at specified level (NULL if unavailable)
`measures`	Named vector of goodness-of-fit statistics (KS statistic, p-value, AIC, BIC)
`convergence`	Integer convergence code (0 = successful)
`method`	Character string of estimation method used
`ci_method`	Character string of confidence interval method used
`ci_level`	Numeric confidence level used
`note`	Character string with interpretation note for KS test

See Also

`fit_dist`

Examples

```
# Simulate tree diameter data
set.seed(123)
diameters <- rweibull3(100, shape = 2.5, scale = 25, loc = 10)

# Fit with confidence bands
fit1 <- fit_weibull_plot(diameters, amplitude = 5, show_ci = TRUE)

# Fit without confidence bands
fit2 <- fit_weibull_plot(diameters, amplitude = 5, show_ci = FALSE)

# Use bootstrap confidence bands
fit3 <- fit_weibull_plot(diameters,
  amplitude = 5,
  ci_method = "bootstrap", n_boot = 500
)
```

`girard` *Girard Form Class Girard Form Class is a form quotient used to estimate taper.*

Description

Girard Form Class Girard Form Class is a form quotient used to estimate taper.

Usage

```
girard(dbh, dbhIn)
```

Arguments

`dbh`	numeric, diameter outside bark at breast height.
`dbhIn`	numeric, diameter inside bark at the top of the first log

References

Strimbu, B. (2021). Dendrometry Field Manual.

`green` *Index of Green*

Description

Index of Green

Usage

```
green(density)
```

Arguments

`density`	numeric, vector of the density.

Value

Index of Green.

`height` *Height of Tree or any vertical Object*

Description

Computes the height of tree, pillar, girder, mast or any vertical object. It allows either slope (in percent) or angle (in degrees or radians). No matter the relative position of the persons who measures the angle or the slope.

Usage

```
height(
  distance,
  top,
  bottom,
  type = c("angle", "slope"),
  angleUnit = c("deg", "rad")
)
```

Arguments

`distance`	numeric, vector of the horizontal distance between object and the person who measures angle.
`top`, `bottom`	numeric vector of top angle and bottom angle respectively (readings from a clinometer).
`type`	the type of `top` and `bottom` measures. Either `"angle"` or `"slope"`. Default is `"slope"`.
`angleUnit`	the unit of `top` and `bottom` measures when `type` = `"angle"`. Either `"deg"` for degree or `"rad"` for radian. Default is `"deg"`.

Value

A vector of heights.

Examples

```
height(10, 80, 17)
height(17, top = -18, bottom = -113)
height(distance = 18, top = 42, bottom = -12, type = "angle", angleUnit = "deg")
height(
  distance = 18:21, top = 42:45, bottom = -12:-15, type = "angle",
  angleUnit = "deg"
)
## Below shows warning messages
height(
  distance = 18:21, top = -42:-45, bottom = -12:-15, type = "angle",
  angleUnit = "deg"
)
```

`Logging` *Tree metrics for logging*

Description

Data frame of 24 rows and 8 columns containing tree measures.

Usage

```
data(Logging)
```

Format

Data frame with twenty five observations and eight variables:

tree	Tree name (scientific gender).
hauteur	Stem length in meter (m).
diametreMedian	Tree median diameter in centimeter (cm).
perimetreMedian	Tree median circumference in centimeter (cm).
diametreSection	Tree diameter at the end in centimeter (cm).
perimetreSection	Tree circumference at the end in centimeter (cm).
diametreBase	Tree diameter at the base in centimeter (cm).
perimetreBase	Tree circumference at the base in centimeter (cm).

Author(s)

Narcisse Yehouenou `<narcisstar211@gmail.com>`

Source

Fake data simulated for tutorial purposes.

Examples

```
# demo(volume)
```

`logLik.fit_dist` *Extract Log-Likelihood*

Description

S3 method to extract the log-likelihood value from a fitted distribution.

Usage

```
## S3 method for class 'fit_dist'
logLik(object, ...)
```

Arguments

`object`	an object of class "fit_dist".
`...`	additional arguments (not used).

Value

Log-likelihood value.

`loreyHeight` *Lorey's mean height*

Description

The average height of the trees in a plot, weighted by their basal area.

Usage

```
loreyHeight(basal, height)
```

Arguments

`basal`	numeric, vector of trees' individual basal area.
`height`	numeric, vector of trees' individual height.

Value

Average Lorey height of a stand.

See Also

`height`, `basal_i`

Examples

```
set.seed(1)
donnee <- data.frame(
  hauteur = rnorm(10, 12, 3),
  area = basal_i(rnorm(10, 100, 20))
)
loreyHeight(basal = donnee$area, height = donnee$hauteur)
```

`makedata` *Create nested data subsets*

Description

A convenient wrapper around `nestedFunBuilder` that creates nested data subsets without applying any function. This is useful for exploring data structure or preparing data for further analysis.

Usage

```
makedata(data, ...)
```

Arguments

`data`	a data frame, list, tibble or object coercible by `as.data.frame` to a data frame containing the variables whose names are given in the factor arguments.
`...`	character strings specifying the names of grouping variables (factors) in `data`. The nesting order follows the argument order.

Details

This function is equivalent to calling `nestedFunBuilder` with `.fun = identity`. It provides a simpler interface when you only need to create nested data structures without applying functions.

Value

A nested list structure where each level corresponds to a factor level, with the deepest level containing the actual data subsets. If no valid factors are provided, returns the original data frame with a warning.

See Also

`nestedFunBuilder` for applying functions to subsets

Examples

```
## Not run:
# require(BiodiversityR)
# data(ifri, package = "BiodiversityR")
# a1 <- makedata(ifri, "forest", "plotID", "species")
# a2 <- makedata(ifri, "species")
# a3 <- makedata(ifri, "forest", "plotID", "species", "size_class")
# identical(makedata(ifri), ifri)

## End(Not run)
```

`param` *Structural parameters for stands*

Description

Computes various forest stand parameters (basal area, mean diameter, height, etc.) for forest inventory data, with support for grouping by multiple factors and plot-level analysis.

Usage

```
param(
  data,
  ...,
  plot = "",
  DBH = "",
  height = "",
  crown = "",
  area = NULL,
  k = 100,
  kCrown = 1
)
```

Arguments

`data`	a data frame, list, tibble or object coercible by `as.data.frame` to a data frame containing the forest inventory variables.
`...`	additional character strings specifying grouping variables (factors) in `data`. Results will be nested by these factors.
`plot`	optional character, name of the variable containing plot identifiers. If empty (""), all data is treated as a single plot.
`DBH`	optional, character, name of the variable containing diameter at breast height measurements.
`height`	optional, character, name of the variable containing tree height measurements.
`crown`	optional, character, name of the variable containing crown diameter measurements.
`area`	optional, numeric value of plot area, or character name of variable containing plot areas. If NULL, density calculations are omitted.
`k`	numeric, conversion factor for basal area calculation (default: 100).
`kCrown`	numeric, conversion factor for crown basal area calculation (default: 1).

Details

The function supports hierarchical grouping by multiple factors. For example, grouping by species and site will create a nested structure where parameters are calculated for each species within each site.

If plot-level analysis is requested (plot != ""), the function will further subdivide each group by plot and calculate parameters for each plot within each group. Else, it treats the entire dataset as a single plot; area should then be specified accordingly.

Blackman and Green indices are returned if combinations of specified factors contain more than one plot. Otherwise, the right (correct) ones are returned as attributes.

Value

A nested list structure containing calculated parameters for each group. Parameters include:

- `MeanDBH`: Mean diameter at breast height
- `Basal`: Basal area per unit area
- `MeanCrown`: Mean crown diameter
- `BasalCrown`: Crown basal area per unit area
- `Height`: Mean height
- `LoreyHeight`: Lorey’s height (basal area weighted mean height)
- `Density`: Number of trees per unit area

Examples

```
param(
  data = Logging, plot = "tree", DBH = "diametreMedian",
  height = "hauteur", crown = "perimetreBase", area = 0.03, kCrown = 100
)

set.seed(123)
Logging$surperficie <- abs(rnorm(24, mean = 0.03, sd = 0.01))
head(Logging)

param(
  data = Logging, plot = "tree", DBH = "diametreMedian",
  height = "hauteur", crown = "perimetreBase", area = "surperficie", kCrown = 100
)

## Not run:
# Basic usage - single plot
params <- param(forest_data, DBH = "dbh", height = "height", area = 1000)

# Multiple plots
params <- param(forest_data,
  plot = "plot_id", DBH = "dbh",
  height = "height", area = "plot_area"
)

# Grouped analysis
params <- param(forest_data,
  plot = "plot_id", DBH = "dbh",
  height = "height", area = 1000,
  "species", "site", "treatment"
)

# Access specific results
oak_site1 <- params$oak$site1

## End(Not run)
```

`print.angle` *Print Angle*

Description

Method to print angle and returns it invisibly.

Usage

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

Arguments

`x`	an angle object.
`...`	further arguments passed to or from other methods.

`print.fit_dist` *Print Method for Fitted Distributions*

Description

S3 method to print a summary of the fitted distribution.

Usage

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

Arguments

`x`	an object of class "fit_dist".
`...`	additional arguments (not used).

Value

The object invisibly.

`print.slope` *Print Slope*

Description

Method to print slope and returns it invisibly.

Usage

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

Arguments

`x`	a slope object.
`...`	further arguments passed to or from other methods.

`reducecoef` *The reduction coefficient*

Description

The reduction coefficient is the ratio between the difference in size at breast height and mid-height on the one hand, and the size at breast height on the other. It is thus the complement to 1 of the coefficient of decrease.

Usage

```
reducecoef(middle, breast)
```

Arguments

`middle`	numeric, the diameter or circumference at middle height.
`breast`	numeric, the diameter or circumference at breast height.

Details

Both `middle` and `breast` arguments should be of the same type (either diameter or circumference). Not mixture.

Value

The reduction coefficient.

See Also

`decrease`

Examples

```
reducecoef(30, 120)
reducecoef(middle = 40, breast = 90)
```

`rfreq` *Relative Frequency*

Description

Relative Frequency in percentage.

Usage

```
rfreq(x)
```

Arguments

`x`	numeric vector.

`sampleSize` *Sample size*

Description

Sample size

Usage

```
sampleSize(
  confLev = 0.95,
  popPro = 0.5,
  errorMargin = 0.05,
  size = NULL,
  method = "",
  cv = NULL
)
```

Arguments

`confLev`	numeric, the confidence level. Default is `0.05`.
`popPro`	numeric, proportion of population which have considered factor. Default is `0.5`.
`errorMargin`	numeric, margin error. Default is `0.05`.
`size`	integer, population size when it is known. If not specified, simple random sampling will be used. Allows infinite.
`method`	optional character string specifying method to use if not simple adjusted is desired. Only "cauchran" is implemented now.
`cv`	variation coefficient.

Value

The sample size.

Note

Population size to be considered as large or infinite heavily depends on error margin. Lower error margin increases population size to be conidered as large or infinite. For errorMargin = .05, size = 152 231 and cauchran 151 760 when confLev = .05

Examples

```
sampleSize(confLev = .95, popPro = 0.4, errorMargin = .05)
sampleSize(confLev = .95, popPro = 0.5, errorMargin = .05, size = 150)
sampleSize(
  confLev = .95, popPro = 0.5, errorMargin = .05, size = 150,
  method = "cauchran"
)
sampleSize()
```

`shape` *The shape coefficient*

Description

The shape coefficient of the tree is the ratio of the actual volume of the tree to the volume of a cylinder having as base the surface of the section at 1.3 m (or a given breast height) and as length, the height (at bole level) of the tree.

Usage

```
shape(volume, height, dbh, basal = NULL)
```

Arguments

`volume`	numeric, tree real volume.
`height`	numeric, tree height.
`dbh`	numeric, diameter at breast height (DBH).
`basal`	numeric, basal area. Is used when `dbh` is not specified.

Value

The shape coefficient.

See Also

`volume`, for tree real volume.

Examples

```
shape(volume = 10000, 11, dbh = 40)
shape(volume = 10000, 11, 40)
shape(volume = 10000, 11, basal = 2256.637)
## Bellow gives warning
shape(volume = 10000, height = 11, dbh = 40, basal = 2256.637)
```

`skewness` *Skewness coefficient*

Description

Skewness coefficient

Usage

```
skewness(x)
```

Arguments

`x`	numeric vector.

Value

The skewness coefficient.

Examples

```
data("Logging")
skewness(Logging$hauteur)
hist(Logging$hauteur, 3)
```

`spNmReduce` *Abbreviates a Botanical or Zoological Latin Name into an Eight-character from 'Gender epithet' to 'G. epithet'*

Description

To abbreviate species name from 'Gender epithet' to 'G. epithet'. Useful in plots with species names.

Usage

```
spNmReduce(name, sep = " ")
```

Arguments

`name`	a factor coercible vector of species name in forms 'Gender epithet'.
`sep`	character string which separates Gender and epithet. Default is space " ".

Details

Returned reduced names are made unique.

Value

A factor vector of species reduced names in forms 'G. epithet'.

See Also

`make.cepnames` in `vegan` package.

`stacking` *Stack all vectors of a data frame or list*

Description

Stacking all columns of a data frame or vectors of a list into a single vector.

Usage

```
stacking(data)
```

Arguments

`data`	data frame, tibble or list.

Value

A vector of all element of the argument `data`.

`summary.fit_dist` *Summary Method for Fitted Distributions*

Description

S3 method to provide a detailed summary of the fitted distribution.

Usage

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

Arguments

`object`	an object of class "fit_dist".
`...`	additional arguments passed to `print.fit_dist`.

Value

The object invisibly.

`Tree` *Dendrometric measures on tree*

Description

Data frame of 10 rows and 5 columns containing tree measures.

Usage

```
data(Tree)
```

Format

Data frame with ten observations and five variables:

circum Tree circumference in centimeter (cm).

dist Horizontal distance between the person measuring angles and the tree (m).

up Angle measured for the top part of the tree in degree (°). It is used to calculate the total tree height.

down Angle measured for the bottom part of the tree in degree (°).

fut Bole angle measure in degree (°); Bole is where the first branch occurs on the trunk. It is used to calculate the merchantable tree height.

Author(s)

Narcisse Yehouenou <`narcisstar211@gmail.com`>

Source

Fake data simulated for tutorial purposes.

Examples

```
# demo(dendro)
```

`vcov.fit_dist` *Extract Variance-Covariance Matrix*

Description

S3 method to extract the variance-covariance matrix from a fitted distribution.

Usage

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

Arguments

`object`	an object of class "fit_dist".
`...`	additional arguments (not used).

Value

Variance-covariance matrix of parameter estimates.

`volume` *Tree stem and log Volume*

Description

Determining the volume of the log or of the tree.

Usage

```
volume(height, dm, do, ds, circum, circumo, circums,
       method = "huber", successive = FALSE, log)
```

Arguments

`height`	numeric, stem (whole bole) length. When `successive` is "`TRUE`", it stands for log length.
`do`, `dm`, `ds`	numeric, respectively base, median and end diameter.
`circumo`, `circum`, `circums`	numeric, respectively base, median and end circumference.
`method`	character string, the method of volume computation. Can be one of "`huber`", "`smalian`", "`cone`", or "`newton`". Default is "`huber`".
`successive`	logical. If `TRUE`, Successive method is applied. is applied. Default is `FALSE`.
`log`	a vector indicating tree to which belongs each log. Is used only when `successive` is "`TRUE`".

Details

Using `method` = cone refers to truncated cone method.

Value

A numeric vector of logs or trees volume.

See Also

`shape`, for shape coefficient.

Examples

```
## huber method
volume(height = 10, dm = 35)
volume(height = 10, circum = 100)

## smalian method
volume(height = 10, do = 45, ds = 15, method = "smalian")
volume(height = 10, circumo = 200, circums = 110, method = "smalian")

## cone method
volume(height = 10, do = 45, ds = 15, method = "cone")
volume(height = 10, circumo = 200, circums = 110, method = "cone")

## newton method
volume(height = 10, dm = 35, do = 45, ds = 15, method = "newton")
volume(
  height = 10, circum = 100, circumo = 200, circums = 110,
  method = "newton"
)
```

Index