Package ‘silviculture’

September 28, 2025

Type Package

Title Utility Functions for Forest Inventory and Silviculture

Version 0.2.0

Description Perform common dendrometry operations such as inventory preparing, and inventory data analysis.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports cli, dplyr, lifecycle, rlang, S7

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://cidree.github.io/silviculture/

Collate 'data.R' 'utils-not-exported.R' 'deprecated-funs.R' 'inventory-summary.R' 'metrics-lidar.R' 'metrics-stand-density.R' 'metrics-stand-level.R' 'metrics-tree-level.R' 'predict-biomass.R' 'predict-height.R' 'sample-size.R' 'silviculture-package.R' 'treatment-thinning.R' 'zzz.R'

NeedsCompilation no

Author Adrián Cidre González [aut, cre],
Aitor Vázquez Veloso [aut]

Maintainer Adrián Cidre González <adrian.cidre@gmail.com>

Repository CRAN

Date/Publication 2025-09-27 23:10:02 UTC

Contents

`biomass_models` *Biomass models*

Description

Biomass models available in `silviculture` package. If you would like to suggest new models, please open a new issue.

Usage

```
biomass_models
```

Format

A `tibble`

`eq_biomass_cudjoe_2024` *Biomass equations for 2 species in Castille and León (Spain)*

Description

Allometric equations adjusted for *Quercus petraea*, and *Pinus sylvestris* in Castille and León (Spain)

Usage

```
eq_biomass_cudjoe_2024(species, component = "AGB", return_rmse = FALSE)
```

Arguments

`species`	A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species.
`component`	A character string specifying the tree component for biomass calculation (e.g., "stem", "branches"). See Details.
`return_rmse`	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

Details

There are three species options in this model:

- **Quercus petraea**
- **Pinus sylvestris**
- **Mixed**: stands with *Quercus petraea* and *Pinus sylvestris*

The tree components include some AGB components:

- **leaves**: only for *P. sylvestris*
- **stem**: for all species
- **medium branches and small brances**: for all species
- **thick branches**: for all species
- **AGB**: total biomass, results of summing the previous components

Value

A S7 list of parameters

See Also

silv_predict_biomass(), biomass_models, eq_biomass_montero_2005(), eq_biomass_dieguez_aranda_2009() eq_biomass_ruiz_peinado_2011(), eq_biomass_ruiz_peinado_2012(), eq_biomass_manrique_2017(), eq_biomass_menendez_2022()

Examples

```
## get model parameters for silv_predict_biomass
eq_biomass_cudjoe_2024("mixed", "AGB")
```

---

eq_biomass_dieguez_aranda_2009 *Biomass equations for Galician species*

Description

Allometric equations adjusted for Galician (Spain) species

Usage

```
eq_biomass_dieguez_aranda_2009(
  species,
  component = "stem",
  return_r2 = FALSE,
  return_rmse = FALSE
)
```

Arguments

`species`	A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species.
`component`	A character string specifying the tree component for biomass calculation (e.g., "tree", "stem", "branches"). See Details.
`return_r2`	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.
`return_rmse`	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

Details

There are seven species included in this model: *Pinus pinaster, Pinaster radiata, Pinus sylvestris, Eucalyptus globulus, Eucalyptus nitens, Quercus robur*, and *Betula alba*

The tree components are divided into groups, and any of them can be introduced in the component argument:

- **AGB**: all aboveground biomass components
- **BGB**: all belowground biomass compoponents
- **tree**: total tree biomass includying AGB and BGB

Then we have the second group of components, which are related to tree groups:

- **stem**: includes the stem and bark
- **branches**: includes all branches
- **roots**: includes the roots (same as BGB)

Finally, we have the last level, which includes tree components (not all of them are available for all species): stem, bark, thick branches (>7cm), medium branches (2-7cm), thin branches (0.5-2cm), twigs (<0.5cm), dry branches, leaves, roots. In some species, there's "stem and thick branches", instead of two groups.

Users can check the list of supported species and their corresponding components in biomass_models.

Value

A S7 list of parameters

See Also

`silv_predict_biomass()`, biomass_models, `eq_biomass_montero_2005()`, `eq_biomass_ruiz_peinado_2011()`, `eq_biomass_ruiz_peinado_2012()`, `eq_biomass_manrique_2017()`, `eq_biomass_menendez_2022()`, `eq_biomass_cudjoe_2024()`

Examples

```
## get model parameters for silv_predict_biomass
eq_biomass_dieguez_aranda_2009("Pinus pinaster", "AGB")
```

eq_biomass_manrique_2017

Biomass equations two Quercus species

Description

Allometric equations adjusted for *Quercus petraea* and *Quercus pyrenaica* in Palencia, Spain

Usage

```
eq_biomass_manrique_2017(
  species,
  component = "AGB",
  return_r2 = FALSE,
  return_rmse = FALSE
)
```

Arguments

`species`	A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species.
`component`	A character string specifying the tree component for biomass calculation (e.g., "stem", "branches"). See Details.
`return_r2`	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.
`return_rmse`	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

Details

There are two species in this model: *Quercus petraea* and *Quercus pyrenaica*

The tree components include:

- **stem**: includes stem and the thickest branches
- **medium branches**
- **thin branches**
- **AGB**: total biomass, results of summing the previous three components

Value

A S7 list of parameters

See Also

silv_predict_biomass(), biomass_models, eq_biomass_montero_2005(), eq_biomass_dieguez_aranda_2009()
eq_biomass_ruiz_peinado_2011(), eq_biomass_ruiz_peinado_2012(), eq_biomass_menendez_2022(),
eq_biomass_cudjoe_2024()

Examples

```
## get model parameters for silv_predict_biomass
eq_biomass_manrique_2017("Quercus petraea", "AGB")
```

---

eq_biomass_menendez_2022

Biomass equations for young Spanish plantations

Description

Allometric equations for young (<30) plantations of 18 Spanish species including broadleaf and conifer species. Only aboveground biomass.

Usage

```
eq_biomass_menendez_2022(species, return_r2 = FALSE, return_rmse = FALSE)
```

Arguments

`species` A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species.

`return_r2` A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

`return_rmse` A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

Details

There are 15 species in this model, including generic equations for *Conifers*, *Deciduous broadleaves*, and *Evergreen broadleaves*.

All the models measure only aboveground biomass.

Value

A S7 list of parameters

See Also

silv_predict_biomass(), biomass_models, eq_biomass_montero_2005(), eq_biomass_dieguez_aranda_2009()
eq_biomass_ruiz_peinado_2011(), eq_biomass_ruiz_peinado_2012(), eq_biomass_manrique_2017(),
eq_biomass_cudjoe_2024()

Examples

```
## get model parameters for silv_predict_biomass
eq_biomass_menendez_2022("Fagus sylvatica")
```

---

eq_biomass_montero_2005

Biomass equations for Spanish species

Description

Allometric equations adjusted for Spanish species

Usage

```
eq_biomass_montero_2005(species, component = "stem", return_r2 = FALSE)
```

Arguments

species	A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species.
component	A character string specifying the tree component for biomass calculation (e.g., "tree", "stem", "branches"). See Details.
return_r2	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

Details

There are 35 species included in the model.

The tree components are divided into groups, and any of them can be introduced in the component argument:

- **AGB**: all aboveground biomass components
- **BGB**: all belowground biomass compoponents
- **tree** or **all**: total tree biomass includying AGB and BGB

Then we have the second group of components, which are related to tree groups:

- **stem**: includes the stem and bark
- **branches**: includes all branches
- **roots**: includes the roots (same as BGB)

Finally, we have the last level, which includes tree components (not all of them are available for all species): stem, bark, thick branches (>7cm), medium branches (2-7cm), thin branches (0.5-2cm), leaves (include needles), roots. In some species, there's "stem and thick branches", instead of two groups.

Users can check the list of supported species and their corresponding components in biomass_models.

Value

A S7 list of parameters

See Also

silv_predict_biomass(), biomass_models, eq_biomass_dieguez_aranda_2009() eq_biomass_ruiz_peinado_2011() eq_biomass_ruiz_peinado_2012(), eq_biomass_manrique_2017(), eq_biomass_menendez_2022(), eq_biomass_cudjoe_2024()

Examples

```
## get model parameters for silv_predict_biomass
eq_biomass_montero_2005("Pinus pinaster", "AGB")
```

eq_biomass_ruiz_peinado_2011 *Biomass equations for Spanish softwood species*

Description

Allometric equations adjusted for Spanish softwood species

Usage

```
eq_biomass_ruiz_peinado_2011(species, component = "stem", return_rmse = FALSE)
```

Arguments

species	A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species.
component	A character string specifying the tree component for biomass calculation (e.g., "tree", "stem", "branches"). See Details.
return_rmse	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

Details

Users can check the list of supported species and their corresponding components in biomass_models.

Value

A S7 list of parameters

See Also

silv_predict_biomass(), biomass_models, eq_biomass_montero_2005(), eq_biomass_dieguez_aranda_2009(), eq_biomass_ruiz_peinado_2012(), eq_biomass_manrique_2017(), eq_biomass_menendez_2022(), eq_biomass_cudjoe_2024()

Examples

```
## get model parameters for silv_predict_biomass
eq_biomass_ruiz_peinado_2011("Pinus pinaster")
```

---

eq_biomass_ruiz_peinado_2012

Biomass equations for Spanish hardwood species

Description

Allometric equations adjusted for Spanish hardwood species

Usage

```
eq_biomass_ruiz_peinado_2012(species, component = "stem", return_rmse = FALSE)
```

Arguments

species	A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species.
component	A character string specifying the tree component for biomass calculation (e.g., "tree", "stem", "branches"). See Details.
return_rmse	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.

Details

Users can check the list of supported species and their corresponding components in biomass_models.

Value

A S7 list of parameters

See Also

silv_predict_biomass(), biomass_models, eq_biomass_montero_2005(), eq_biomass_dieguez_aranda_2009() eq_biomass_ruiz_peinado_2011(), eq_biomass_manrique_2017(), eq_biomass_menendez_2022(), eq_biomass_cudjoe_2024()

Examples

```
## get model parameters for silv_predict_biomass
eq_biomass_ruiz_peinado_2012("Quercus suber")
```

eq_hd_vazquez_veloso_2025

Estimates tree height from DBH

Description

This function is intended to be used in `silv_predict_height()`. It implements the h-d equations developed in Vázquez-Veloso et al. (2025). These equations have been developed using the Spanish National Forest Inventory, and therefore, they should only be applied within Spain. The model includes parameters for 91 tree species.

Usage

```
eq_hd_vazquez_veloso_2025(
  species,
  bioregion = "mediterranean",
  origin = "natural",
  mixture = "pure"
)
```

Arguments

`species` A character string specifying the scientific name of the tree species. It can be a column name if all the species are included in this model. See Details for available species. If not specified, it takes the value "All the species", which corresponds to a generic model applicable to all species.

`bioregion` The biogeopgrahic region of the species. Available options are: `mediterranean`, `atlantic`, `alpine`, and `macaronesian`. If not specified, it takes the value `mediterranean`, which is the most common region in Spain. You can check the distribution of regions here: https://ars.els-cdn.com/content/image/1-s2.0-S037811272500489X-gr1.jpg

`origin` The origin of the stand. Available options are: `natural` and `plantation`. If not specified, it takes the value `natural`, which is the most common origin in Spain.

`mixture` The species available in the stand. Available options are: `pure` and `mix`. Consider the characteristics of the plot you are evaluating and not the entire forest, as the conditions of each stand are different. In this study, it was considered a stand to be mixed when the combined proportion of at least two species exceeds 90% of the plot's basal area, and the proportion of both species is greater than 15% of the total. It does not matter which species is accompanying or the proportion of mixing. If not specified, it takes the value `pure`, which is the most common condition in Spain.

Details

Details...#TODO

Value

A numeric vector with predicted height

References

Vázquez-Veloso, A., Yang, S.-I., Bullock, B.P., Bravo, F., 2025. One model to rule them all: A nationwide height–diameter model for 91 Spanish forest species. Forest Ecology and Management 595, 122981. https://doi.org/10.1016/j.foreco.2025.122981

See Also

`silv_predict_height()`

Examples

```
1 + 1 #TODO
```

`inventory_samples` *Forest inventory samples*

Description

Inventory data from Spanish National Forest Inventory

Usage

```
inventory_samples
```

Format

A `tibble`

lid_fcov *Calculate Forest Fraction Cover from LiDAR Data*

Description

This function calculates the forest fraction cover (Fcov) from LiDAR data. The Fcov in LiDAR is defined as the proportion of first returns above a specified height threshold (default: 5 meters) relative to the total number of first returns.

Usage

```
lid_fcov(z, rn, th = 5)
```

Arguments

z	A numeric vector representing the heights of LiDAR returns
rn	An integer vector indicating the return number for each LiDAR point. First returns are identified by a value of 1
th	a numeric vector of length one specifying the height threshold

Value

A numeric value representing the forest fraction cover, which is the proportion of first returns with heights greater than 5 meters.

Examples

```
# Example data
z <- c(2, 6, 10, 4, 15)
rn <- c(1, 1, 2, 1, 1)

# Calculate forest fraction cover
lid_fcov(z, rn)
```

lid_lhdi *LiDAR-derived Height Diversity Index (LHDI)*

Description

LiDAR metric that calculates the LiDAR Height Diversity Index, which can be used in lidR *_metrics functions

Usage

```
lid_lhdi(z, interval = 0.5)
```

Arguments

`z`	coordinate Z (height) of the point
`interval`	height of the intervals to calculate the metric

Value

numeric

References

Listopad, C. M. C. S., Masters, R. E., Drake, J., Weishampel, J., & Branquinho, C. (2015). Structural diversity indices based on airborne LiDAR as ecological indicators for managing highly dynamic landscapes. Ecological Indicators, 57, 268–279. doi:10.1016/j.ecolind.2015.04.017

Examples

```
1 + 1 ## TODO
```

`plot` *Plot an object*

Description

Generic for plotting objects.

Usage

```
plot(x, ...)
```

Arguments

`x`	Object to plot.
`...`	Other arguments passed to methods.

Value

Usually called for side-effects (producing a plot).

silv_basal_area *Calculates Basal Area*

Description

[Deprecated]

Calculates Basal Area in square meters.

Usage

```
silv_basal_area(diameter, ntrees = NULL, units = "cm")
```

Arguments

diameter	Numeric vector of diameters or diameter classes
ntrees	Numeric vector with number of trees of the diameter class per hectare. If ntrees = NULL, the function will assume that each diameter corresponds to only one tree. Therefore, basal area will be calculated for each individual tree
units	The units of the diameter (one of cm, mm, or m)

Details

The function uses the next formula:

$G = \frac{\pi}{40000} \cdot D^2$

where G is the basal area in m^2, and D is the diameter in the units specified in the function. It is recommended to use the squared mean diameter calculated with silv_sqrmean_diameter

Value

A numeric vector

Examples

```
## calculate G for inventory data grouped by plot_id and species
library(dplyr)
inventory_samples |>
mutate(dclass = silv_diametric_class(diameter)) |>
  summarise(
    height = mean(height, na.rm = TRUE),
    ntrees = n(),
    .by    = c(plot_id, species, dclass)
  ) |>
  mutate(
    ntrees_ha = silv_ntrees_ha(ntrees, plot_size = 10),
    dg        = silv_sqrmean_diameter(dclass, ntrees_ha),
    g         = silv_basal_area(dclass, ntrees_ha),
    .by       = c(plot_id, species)
  )

## calculate individual basal area
silv_basal_area(c(23, 11, 43.5, 94))
```

silv_biomass *Calculate Tree Biomass*

Description

[Deprecated]

Computes the biomass of a tree species using species-specific allometric equations (in kg).

Usage

```
silv_biomass(
  diameter = NULL,
  height = NULL,
  ntrees = NULL,
  species = NULL,
  component = "stem",
  model = "ruiz-peinado-2012",
  return_rmse = FALSE,
  quiet = FALSE
)
```

Arguments

diameter	A numeric vector of tree diameters (in cm).
height	A numeric vector of tree heights (in m).
ntrees	An optional numeric value indicating the number of trees in this diameter-height class. Defaults to 1 if NULL.
species	A character string specifying the scientific name of the tree species. See Details for available species.
component	A character string specifying the tree component for biomass calculation (e.g., "tree", "stem", "branches"). See Details.
model	A character string indicating the ID of the publication in which the model was developed. Currently supported models: "ruiz-peinado-2012" (hardwood species in Spain) and "ruiz-peinado-2011" (softwood species in Spain). See Details.
return_rmse	A logical value. If TRUE, the function returns the root mean squared error (RMSE) of the selected model instead of the biomass value.
quiet	A logical value. If TRUE, suppresses any informational messages.

Details

The function estimates biomass using validated allometric models available in the dataset biomass_models. The available models include:

- **ruiz-peinado-2011**: Developed for softwood species in Spain.
- **ruiz-peinado-2012**: Developed for hardwood species in Spain.

Users can check the list of supported species and their corresponding components in biomass_models.

If you would like to suggest additional models, please open a new issue on GitHub.

Value

A numeric vector of biomass values (in kg). If `return_rmse = TRUE`, returns the RMSE instead.

Examples

```
# Calculate biomass for a single tree
silv_biomass(
  diameter = 45,
  height   = 22,
  species  = "Pinus pinaster",
  model    = "ruiz-peinado-2011"
)
```

`silv_density_hart` *Hart or Hart-Becking spacing index*

Description

Calculates the Hart Index or the Hart-Becking Index for even-aged stands

Usage

```
silv_density_hart(h0, ntrees, which = "hart")
```

Arguments

`h0`	Numeric vector with dominant height
`ntrees`	Numeric vector with number of trees of the dominant height per hectare
`which`	A character with the name of the index (either `hart` or `hart-brecking`). See details

Details

The spacing index can be used to determine whether a thinning is needed or not, and also to determine how intense it should be.

- **Hart Index**: it assumes even-aged stands with square planting pattern.
- **Hart-Brecking Index**: it assumes triangular planting pattern.

Value

A numeric vector

References

Assmann, E. (1970) The principles of forest yield study: Studies in the organic production, structure, increment, and yield of forest stands. Pergamon Press, Oxford.

Examples

```
library(dplyr)
## Calculate spacing index for each plot
inventory_samples |>
  summarise(
    h0     = silv_stand_dominant_height(diameter, height),
    ntrees = n(),
    .by    = plot_id
  ) |>
  ## calculate number of trees per hectare
  mutate(ntrees_ha = silv_density_ntrees_ha(ntrees, plot_size = 14.1)) |>
  mutate(spacing = silv_density_hart(h0, ntrees_ha))
```

`silv_density_ntrees_ha` *Calculates number of trees per hectare*

Description

Calculates number of trees per hectare for a given plot size and shape

Usage

```
silv_density_ntrees_ha(ntrees, plot_size, plot_shape = "circular")
```

Arguments

`ntrees` A numeric vector representing the number of trees in a sampling plot

`plot_size` A numeric vector of length one for circular radius in meters; or a numeric vector of length two for each side of a rectangular plot shape

`plot_shape` The shape of the sampling plot. Either `circular` or `rectangular`

Value

A numeric vector

Examples

```
library(dplyr)
## Circular plot of 10 meters radius
inventory_samples |>
  count(plot_id, species) |>
  mutate(
    ntrees_ha = silv_density_ntrees_ha(n, plot_size = 10)
  )

## Rectangular plot of 10x15 meters
inventory_samples |>
  count(plot_id, species) |>
  mutate(
    ntrees_ha = silv_density_ntrees_ha(
      n,
      plot_size = c(10, 15),
      plot_shape = "rectangular"
     )
  )
```

`silv_density_sdi` *Calculates the Stand Density Index*

Description

The Stand Density Index (SDI) is relationship between the average tree size and density of trees per hectare.

Usage

```
silv_density_sdi(ntrees, dg, classify = FALSE, max_sdi = NULL)
```

Arguments

`ntrees`	A numeric vector representing the number of trees per hectare
`dg`	A numeric vector of quadratic mean diameters
`classify`	whether to classify the values using USDA thresholds
`max_sdi`	used when `classify = TRUE`. The maximum SDi, which depends on the species, stand type, and site

Details

The SDI has different interpretation depending on the species, location, and also the management type (even-aged, uneven-aged...). The value of maximum SDI must be determined from the literature and used carefully. The option `classify = TRUE` will use this value to classify the SDI in low density (<24%), moderate density (24-35%), high density (34-55%), and extremely high density (>55%).

Value

A numeric vector

Examples

```
## calculate SDI for a Pinus sulvestris stand (max 990)
silv_density_sdi(ntrees = 800, dg = 23.4, max_sdi = 990)

## check base classification (other can be used)
silv_density_sdi(ntrees = 800, dg = 23.4, classify = TRUE, max_sdi = 990)
```

`silv_diametric_class` *Classify diameters in classes*

Description

[Deprecated]

Classifies the measured diameters into classes of a specified length

Usage

```
silv_diametric_class(
  diameter,
  dmin = 7.5,
  dmax = NULL,
  class_length = 5,
  include_lowest = TRUE,
  return_intervals = FALSE
)
```

Arguments

`diameter` A numeric vector of diameters

`dmin` The minimum inventory diameter in centimeters

`dmax` The maximum inventory diameter in centimeters. Values that are greater than `dmax` are included in the greatest class

`class_length` The length of the class in centimeters

`include_lowest` Logical. If TRUE (the default), the intervals are `[dim1, dim2)`. If FALSE, the intervals are `(dim1, dim2]`

[dim1, dim2)`. If FALSE, the intervals are` (dim1, dim2]: R:dim1,%20dim2)%60.%20If%20FALSE

`return_intervals` If FALSE, it returns the intervals. Otherwise (the default), it returns the class center

Value

A numeric vector

Examples

```
library(dplyr)
inventory_samples |>
  mutate(dclass = silv_diametric_class(diameter))
```

`silv_dominant_height` *Calculates the dominant height*

Description

[Deprecated]

Calculates the dominant height using the Assman equation or the Hart equation

Usage

```
silv_dominant_height(diameter, height, ntrees = NULL, which = "assman")
```

Arguments

`diameter`	Numeric vector with diameter classes
`height`	Numeric vector with averaged heights by diameter class
`ntrees`	Optional. Numeric vector with number of trees per hectare. Use this argument when you have aggregated data by diametric classes (see details).
`which`	The method to calculate the dominant height (see details)

Details

The dominant height H_0 is the mean height of dominant trees, which is less affected than overall mean height by thinning or other treatments.

- **Assman**: calculates the H_0 as the mean height of the 100 thickest trees per hectare
- **Hart**: calculates the H_0 as the mean height of the 100 tallest trees per hectare

When `ntrees = NULL`, the function will assume that each diameter and height belongs to only one tree. If you have data aggregated by hectare, you'll use the number of trees per hectare in this argument.

Value

A numeric vector

References

Assmann, E. (1970) The principles of forest yield study: Studies in the organic production, structure, increment, and yield of forest stands. Pergamon Press, Oxford.

Examples

```
## calculate h0 for inventory data grouped by plot_id and species
library(dplyr)
inventory_samples |>
mutate(dclass = silv_diametric_class(diameter)) |>
  summarise(
    height = mean(height, na.rm = TRUE),
    ntrees = n(),
    .by    = c(plot_id, species, dclass)
  ) |>
  mutate(
    ntrees_ha = silv_ntrees_ha(ntrees, plot_size = 10),
    h0        = silv_dominant_height(dclass, height, ntrees_ha),
    .by       = c(plot_id, species)
  )
```

`silv_lorey_height` *Calculates Lorey's Height*

Description

#' @description **[Deprecated]**

Usage

```
silv_lorey_height(height, g, ntrees = NULL)
```

Arguments

`height`	Numeric vector of heights
`g`	Numeric vector of basal areas
`ntrees`	Optional. Numeric vector of number of trees per hectare. Use this argument when you have aggregated data by diametric classes (see details).

Details

Tree's mean height weighted by basal area

The function calculates Lorey's mean height according to:

$$h_L = \frac{\sum n_i g_i h_i}{\sum n_i g_i}$$

When ntrees is not provided (i.e. `ntrees = NULL`) the formula is simply the weighted mean of the provided heights and basal areas:

$$h_L = \frac{\sum g_i h_i}{\sum g_i}$$

Value

A numeric vector

Examples

```
## Calculate Lorey's Height by plot and species
library(dplyr)
inventory_samples |>
  mutate(g = silv_basal_area(diameter)) |>
  summarise(
    lh  = silv_lorey_height(height, g),
    .by = c(plot_id, species)
  )
```

`silv_ntrees_ha` *Calculates number of trees per hectare*

Description

[Deprecated] Calculates number of trees per hectare for a given plot size and shape

Usage

```
silv_ntrees_ha(ntrees, plot_size, plot_shape = "circular")
```

Arguments

`ntrees`	A numeric vector representing the number of trees in a sampling plot
`plot_size`	A numeric vector of length one for circular radius in meters; or a numeric vector of length two for each side of a rectangular plot shape
`plot_shape`	The shape of the sampling plot. Either `circular` or `rectangular`

Value

A numeric vector

Examples

```
library(dplyr)
## Circular plot of 10 meters radius
inventory_samples |>
  count(plot_id, species) |>
  mutate(
    ntrees_ha = silv_ntrees_ha(n, plot_size = 10)
  )

## Rectangular plot of 10x15 meters
inventory_samples |>
  count(plot_id, species) |>
  mutate(
    ntrees_ha = silv_ntrees_ha(
      n,
      plot_size = c(10, 15),
      plot_shape = "rectangular"
     )
  )
```

---

silv_predict_biomass *Calculate Tree Biomass*

Description

Computes the biomass of a tree species using species-specific allometric equations (in kg). Currently, only equations for Spain are available.

Usage

```
silv_predict_biomass(
  diameter = NULL,
  height = NULL,
  model,
  ntrees = NULL,
  quiet = FALSE
)
```

Arguments

`diameter` A numeric vector of tree diameters (in cm).

`height` A numeric vector of tree heights (in m).

`model` A function. A function with the structure `eq_biomass_*()` with additional arguments depending on the model used.

`ntrees` An optional numeric value indicating the number of trees in this diameter-height class. Defaults to 1 if `NULL`.

`quiet` A logical value. If `TRUE`, suppresses any informational messages.

Details

The function estimates biomass using validated allometric models available in the dataset biomass_models. The available models include:

- `eq_biomass_ruiz_peinado_2011()`: Developed for softwood species in Spain.
- `eq_biomass_ruiz_peinado_2012()`: Developed for hardwood species in Spain.

Users can check the list of supported species and their corresponding components in biomass_models.

If you would like to suggest additional models, please open a new issue on GitHub.

Value

A numeric vector

See Also

biomass_models, `eq_biomass_montero_2005()`, `eq_biomass_dieguez_aranda_2009()`, `eq_biomass_ruiz_peinado_20`
`eq_biomass_ruiz_peinado_2012()`, `eq_biomass_manrique_2017()`, `eq_biomass_menendez_2022()`, `eq_biomass_cudjoe_2024()`

Examples

```
# Calculate biomass for a single tree
silv_predict_biomass(
  diameter = 45,
  height   = 22,
  model    = eq_biomass_ruiz_peinado_2011("Pinus pinaster")
)
```

`silv_predict_height` *Estimates tree height from DBH*

Description

Estimates total tree height using height-diameter (h-d) equations. Currently, only models developed for Spain are available.

Usage

```
silv_predict_height(diameter, model, quiet = FALSE)
```

Arguments

`diameter` Numeric vector with diameters in cm

`model` A function. A function with the structure `eq_hd_*()` with additional arguments depending on the specific model. Currently only `eq_hd_vazquez_veloso_2025()` is available.

`quiet` A logical value. If TRUE, suppresses any informational messages.

Details

The function estimates total tree height (in meters) using diameter at breast height (in centimeters), and may require additional information depending on the specific model. See each model's documentation for details.

Value

A numeric vector with predicted heights

References

References for the models available:

- `eq_hd_vazquez_veloso_2025()`: Vázquez-Veloso, A., Yang, S.-I., Bullock, B.P., Bravo, F., 2025. One model to rule them all: A nationwide height–diameter model for 91 Spanish forest species. Forest Ecology and Management 595, 122981. https://doi.org/10.1016/j.foreco.2025.122981

See Also

`eq_hd_vazquez_veloso_2025()`

Examples

```
1 + 1 #TODO
```

`silv_sample_size` *Calculates sample size for a random sampling inventory*

Description

[Deprecated]

Usage

```
silv_sample_size(
  x,
  plot_size,
  total_area,
  method = "random",
  max_error = 0.05,
  conf_level = 0.95,
  max_iter = 1000,
  quiet = FALSE
)
```

Arguments

`x`	vector of field survey
`plot_size`	a numeric vector of length one with plot size in squared meters
`total_area`	total area of the study area in squared meters
`method`	sampling method. Available options are `random`
`max_error`	maximum allowed error
`conf_level`	confidence level
`max_iter`	maximum number of iteration to find the plot size
`quiet`	if `TRUE`, messages will be supressed

Value

SampleSize object

Examples

```
## pilot inventory measuring 4 plots of 25x25 meters
## total forest area 15 ha
## measured variable (x): basal area per hectare
silv_sample_size(
  x          = c(33, 37.5, 42, 35.2),
  plot_size  = 25 * 25,  # squared plot of 25x25
  total_area = 15 * 1e4, # 15 ha
  max_error  = 0.05,
  conf_level = 0.95,
  max_iter   = 100
)
```

`silv_sample_size_simple` *Calculates sample size for a simple random sampling (SRS)*

Description

Calculates the sample size needed for a SRS inventory, estimated from pilot inventory data.

Usage

```
silv_sample_size_simple(
  x,
  plot_size,
  total_area,
  max_error = 0.05,
  conf_level = 0.95,
  max_iter = 1000,
  quiet = FALSE
)
```

Arguments

`x`	vector of the variable measured in the pilot inventory (e.g. basal area, volume)
`plot_size`	a numeric vector of length one with plot size in squared meters
`total_area`	total area of the study area in squared meters
`max_error`	maximum allowed relative error
`conf_level`	confidence level
`max_iter`	maximum number of iteration to find the plot size
`quiet`	if `TRUE`, messages will be supressed

Details

Sample size is very important to be optimized, since a small sample size will entail a higher error, while a huge sample size will entail higher costs. The SRS is typically used for random sampling, although it might be used also for regular sampling. The number of samples is calculated using the expression:

$n \geq \frac{t^2 \cdot CV^2}{\epsilon^2 + \frac{t^2 \cdot CV^2}{N}}$

Where:

- **t**: the value of student's t for given sample size of the pilot inventory
- **CV**: the coefficient of variation of `x`
- ϵ: the relative error (`max_error`)
- **N**: the size of the pilot inventory

`x` is a variable measured in a pilot inventory. Let's say we measure forest variables in 10 pilot plots, aiming at basal area measurement so we have to measure only the DBH. After some calculations, we will have the basal area per hectare in each of the 10 plots. The sample size is then calculated from the variation of these values and the error that we will allow.

Value

SimpleSampleSize object

Examples

```
## pilot inventory measuring 4 plots of 25x25 meters
## total forest area 15 ha
## measured variable (x): basal area per hectare
silv_sample_size_simple(
  x          = c(33, 37.5, 42, 35.2),
  plot_size  = 25 * 25,  # squared plot of 25x25
  total_area = 15 * 1e4, # 15 ha
  max_error  = 0.05,
  conf_level = 0.95,
  max_iter   = 100
)
```

`silv_sample_size_stratified` *Calculates sample size for a stratified sampling*

Description

Calculates the sample size needed for a stratified inventory, estimated from pilot inventory data.

Usage

```
silv_sample_size_stratified(
  data,
  x,
  strata,
  total_area,
  plot_size,
  method = "optimal",
  cost = NA,
  max_error = 0.05,
  conf_level = 0.95,
  max_iter = 1000,
  currency = "EUR",
  quiet = FALSE
)
```

Arguments

`data` a `data.frame` of pilot inventory data

`x` name of the variable in `data` that was measured (e.g. basal area, volume)

`strata` name of the variable in `data` with the name of the stratum

`total_area` name of the variable in `data` with the area of the stratum

`plot_size` a numeric vector of length one with plot size in squared meters

`method` a charater vector of length one with the id of the method. Available options are `optimal`, `cost`, and `prop`. See details

`cost` name of the variable in `data` with the average cost of measuring one plot of the stratum. Used with `method = 'cost'` for sample size, and for message output in other methods

`max_error` maximum allowed relative error

`conf_level` confidence level

`max_iter` maximum number of iteration to find the plot size

`currency` currency to be shown in console output when using `method = 'cost'`

`quiet` if `TRUE`, messages will be supressed

Details

Stratified Sampling calculates the number of plots to be inventored in different strata. For instance, you might have *Pinus sylvestris* and *Pinus pinaster* plots in the same forest, and you might want to get the optimal number of plots for field inventory of each stratum, for a given maximum relative error (e.g. 5%), and with a certain level of confidence (e.g 95%). Of course, the area of *P. sylvestris* will be different than the area occupied by *P. pinaster*. For instance, the total area of *P. sylvestris* could be 100 ha, while the area of *P. pinaster* could be 200 ha. Therefore, you need to create a pilot inventory and measure a variable such as basal area maybe in 5 pilot plots of *P. sylvestris* and 7 pilot plots of *P. pinaster*. With that data collected, you can use three stratified sample size methods:

- **Optimal Allocation with Constant Cost**: using `method = 'optimal'`. The sampling units are distributed within the different strata taking into account the size (e.g. 100 ha vs 200 ha) and the heterogeinity (e.g. differences in basal area). It minimizes the number of sampling units.

$$n = \frac{t_{n-m}^2 \cdot (\sum_{j=1}^{j=m} P_j \cdot s_j)^2}{\epsilon^2 + \frac{t_{n-m}^2 \cdot \sum_{j=1}^{j=m} P_j \cdot s_j^2}{N}}$$

- **Optimal Allocation with Variable Cost**: using `method = 'cost'`. This method needs to know the cost of a sampling unit in each strata. It will minimize the cost of the inventory, taking into account the size, the heterogeinity, and the cost of the sampling unit of the strata.

$$n = \frac{t_{n-m}^2 \cdot (\sum_{j=1}^{j=m} \cdot P_j \cdot s_j \cdot \sqrt{c_j}) \cdot (\sum_{j=1}^{j=m} \cdot \frac{P_j \cdot s_j}{\sqrt{c_j}})}{\epsilon^2 + \frac{t_{n-m}^2 \cdot \sum_{j=1}^{j=m} P_j \cdot s_j^2}{N}}$$

- **Proportional Allocation**: using `method = 'prop'`. The sampling units are distributed proportional to the size of the strata. In the example, 33% of the estimated sampling units will be allocated to *P. sylvestris* and 66% to *P. pinaster*.

$$n = \frac{t_{n-m}^2 \cdot \sum_{j=1}^{j=m} P_j \cdot s_j^2}{\epsilon^2 + \frac{t_{n-m}^2 \cdot \sum_{j=1}^{j=m} P_j \cdot s_j^2}{N}}$$

Where:

- **n**: estimated sample size
- **t**: the value of student's t
- P_j: proportion of pilot plots of j^{th} strata
- s_j: standard deviation of `x`
- s_j^2: variance of `x`
- **N**: population size (number of plots of `plot_size` that fit in `total_area`)
- ϵ: maximum allowed absolute error. Calculated from `x` and `max_error`
- **N**: the size of the pilot inventory

Value

S7 `StratifiedSampleSize` object with:

- **results**: `data.frame` with the main results by stratum

- **strata_error**: `data.frame` with maximum absolute error ∓ C.I (max_abs_error, x_min, x_max), and the esimator of the typical error ∓ C.I (sampling error, x_ci_lo, x_ci_hi)
- **sampling_error**: `data.frame` with the maximum absolute error ∓ C.I (max_abs_error, x_min, x_max), and the typical sampling error of the weighted mean ∓ C.I (sampling error, x_ci_lo, x_ci_hi)
- **sampling_opts**: `list` with function options

Examples

```
## read pilot inventory ficticious data
data_path <- system.file("extdata/pilot_inventory.csv", package = "silviculture")
inventory_tbl <- read.csv(data_path)

## calculate sample size
sample_size_list <- silv_sample_size_stratified(
  data  = inventory_tbl,
  x     = basal_area,
  strata = stratum,
  total_area = area,
  method = "optimal",
  cost = cost,
  plot_size = 100,
  conf_level = .95,
  max_error = .05
)
```

`silv_spacing_index` *Hart or Hart-Becking spacing index*

Description

[Deprecated]

Calculates the Hart Index or the Hart-Becking Index for even-aged stands

Usage

```
silv_spacing_index(h0, ntrees, which = "hart")
```

Arguments

`h0`	Numeric vector with dominant height
`ntrees`	Numeric vector with number of trees of the dominant height per hectare
`which`	A character with the name of the index (either `hart` or `hart-brecking`). See details

Details

The spacing index can be used to determine whether a thinning is needed or not, and also to determine how intense it should be.

- **Hart Index**: it assumes even-aged stands with square planting pattern.
- **Hart-Brecking Index**: it assumes triangular planting pattern.

Value

A numeric vector

References

Assmann, E. (1970) The principles of forest yield study: Studies in the organic production, structure, increment, and yield of forest stands. Pergamon Press, Oxford.

Examples

```
library(dplyr)
## Calculate spacing index for each plot
inventory_samples |>
  summarise(
    h0     = silv_dominant_height(diameter, height),
    ntrees = n(),
    .by    = plot_id
  ) |>
  ## calculate number of trees per hectare
  mutate(ntrees_ha = silv_ntrees_ha(ntrees, plot_size = 14.1)) |>
  mutate(spacing = silv_spacing_index(h0, ntrees_ha))
```

`silv_sqrmean_diameter` *Calculates the quadratic mean diameter (QMD)*

Description

[Deprecated]

Usage

```
silv_sqrmean_diameter(diameter, ntrees = NULL)
```

Arguments

`diameter` Numeric vector of diameters or diameter classes

`ntrees` Numeric vector with number of trees of the diameter class per hectare. If `ntrees = NULL`, the function will assume that each diameter corresponds to only one tree. Therefore, the QMD will be calculated for each individual tree

Value

A numeric vector

Examples

```
## calculate dg for inventory data grouped by plot_id and species
library(dplyr)
inventory_samples |>
mutate(dclass = silv_diametric_class(diameter)) |>
  summarise(
    height = mean(height, na.rm = TRUE),
    ntrees = n(),
    .by    = c(plot_id, species, dclass)
  ) |>
  mutate(
    ntrees_ha = silv_ntrees_ha(ntrees, plot_size = 10),
    h0        = silv_dominant_height(dclass, height, ntrees_ha),
    dg        = silv_sqrmean_diameter(dclass, ntrees_ha),
    .by       = c(plot_id, species)
  )

## calculate dg for a vector of diameters
silv_sqrmean_diameter(c(12.5, 23.5, 14, 16, 18.5))
```

`silv_stand_basal_area` *Calculates Basal Area*

Description

Calculates Basal Area in square meters.

Usage

```
silv_stand_basal_area(diameter, ntrees = NULL, units = "cm")
```

Arguments

`diameter` Numeric vector of diameters or diameter classes

`ntrees` Numeric vector with number of trees of the diameter class per hectare. If `ntrees = NULL`, the function will assume that each diameter corresponds to only one tree

`units` The units of the diameter (one of `mm`, `cm`, `dm`, or `m`)

Details

The function uses the next formula:

$G = \frac{\pi}{40000} \cdot D^2 \cdot \text{ntrees}$

where G is the basal area in m^2, and D is the diameter in cm. If ntrees in the number of trees per hectare, then the result will be m^2/ha. It is recommended to use the squared mean diameter calculated with silv_stand_qmean_diameter().

Note that if ntrees = NULL, the output of the function will be exactly the same as in silv_tree_basal_area().

Value

A numeric vector

Examples

```
## calculate G for inventory data grouped by plot_id and species
library(dplyr)
inventory_samples |>
mutate(dclass = silv_tree_dclass(diameter)) |>
  summarise(
    height = mean(height, na.rm = TRUE),
    ntrees = n(),
    .by    = c(plot_id, species, dclass)
  ) |>
  mutate(
    ntrees_ha = silv_density_ntrees_ha(ntrees, plot_size = 10),
    dg        = silv_stand_qmean_diameter(dclass, ntrees_ha),
    g         = silv_stand_basal_area(dclass, ntrees_ha),
    .by       = c(plot_id, species)
  )
```

silv_stand_dominant_diameter

Calculates the dominant diameter

Description

Calculates the dominant diameter using Assman and Friedrich method, or Weise method

Usage

```
silv_stand_dominant_diameter(
  diameter,
  ntrees = NULL,
  which = "assman",
  quiet = FALSE
)
```

Arguments

`diameter` Numeric vector with diameter classes

`ntrees` Optional. Numeric vector with number of trees per hectare. Use this argument when you have aggregated data by diametric classes (see details).

`which` The method to calculate the dominant diameter (see details)

`quiet` if `TRUE`, messages will be supressed

Details

The dominant diameter D_0 is the mean diameter of the 100 thickest trees per hectare. Therefore, `diameter` and `ntrees` should be vectors of the same length.

- **Assman**: calculates the D_0 as the mean diameter of the 100 thickest trees per hectare
- **Weise**: calculates the D_0 as the quadratic mean diameter of the 20% thickest trees per hectare

Value

A numeric vector

Examples

```
## calculate d0 for inventory data grouped by plot_id and species
library(dplyr)
inventory_samples |>
mutate(dclass = silv_tree_dclass(diameter)) |>
  summarise(
    height = mean(height, na.rm = TRUE),
    ntrees = n(),
    .by    = c(plot_id, species, dclass)
  ) |>
  mutate(
    ntrees_ha = silv_density_ntrees_ha(ntrees, plot_size = 10),
    d0        = silv_stand_dominant_diameter(dclass, ntrees_ha),
    .by       = c(plot_id, species)
  )
```

`silv_stand_dominant_height`

Calculates the dominant height

Description

Calculates the dominant height using the Assman equation or the Hart equation

Usage

```
silv_stand_dominant_height(diameter, height, ntrees = NULL, which = "assman")
```

Arguments

`diameter` Numeric vector with diameter classes

`height` Numeric vector with averaged heights by diameter class

`ntrees` Optional. Numeric vector with number of trees per hectare. Use this argument when you have aggregated data by diametric classes (see details).

`which` The method to calculate the dominant height (see details)

Details

The dominant height H_0 is the mean height of dominant trees, which is less affected than overall mean height by thinning or other treatments.

- **Assman**: calculates the H_0 as the mean height of the 100 thickest trees per hectare
- **Hart**: calculates the H_0 as the mean height of the 100 tallest trees per hectare

When `ntrees = NULL`, the function will assume that each diameter and height belongs to only one tree. If you have data aggregated by hectare, you'll use the number of trees per hectare in this argument.

Value

A numeric vector

References

Assmann, E. (1970) The principles of forest yield study: Studies in the organic production, structure, increment, and yield of forest stands. Pergamon Press, Oxford.

Examples

```
## calculate h0 for inventory data grouped by plot_id and species
library(dplyr)
inventory_samples |>
mutate(dclass = silv_tree_dclass(diameter)) |>
  summarise(
    height = mean(height, na.rm = TRUE),
    ntrees = n(),
    .by    = c(plot_id, species, dclass)
  ) |>
  mutate(
    ntrees_ha = silv_density_ntrees_ha(ntrees, plot_size = 10),
    h0        = silv_stand_dominant_height(dclass, height, ntrees_ha),
    .by       = c(plot_id, species)
  )
```

`silv_stand_lorey_height` *Calculates Lorey's Height*

Description

Tree's mean height weighted by basal area

Usage

```
silv_stand_lorey_height(height, g, ntrees = NULL)
```

Arguments

`height`	Numeric vector of heights
`g`	Numeric vector of basal areas
`ntrees`	Optional. Numeric vector of number of trees per hectare. Use this argument when you have aggregated data by diametric classes (see details).

Details

The function calculates Lorey's mean height according to:

$$h_L = \frac{\sum n_i g_i h_i}{\sum n_i g_i}$$

When ntrees is not provided (i.e. `ntrees` = `NULL`) the formula is simply the weighted mean of the provided heights and basal areas:

$$h_L = \frac{\sum g_i h_i}{\sum g_i}$$

Value

A numeric vector

Examples

```
## Calculate Lorey's Height by plot and species
library(dplyr)
inventory_samples |>
  mutate(g = silv_tree_basal_area(diameter)) |>
  summarise(
    lh  = silv_stand_lorey_height(height, g),
    .by = c(plot_id, species)
  )
```

silv_stand_qmean_diameter

Calculates the quadratic mean diameter (QMD)

Description

Calculates the quadratic mean diameter (QMD)

Usage

```
silv_stand_qmean_diameter(diameter, ntrees = NULL)
```

Arguments

`diameter`	Numeric vector of diameters or diameter classes
`ntrees`	Numeric vector with number of trees of the diameter class per hectare. If `ntrees = NULL`, the function will assume that each diameter corresponds to only one tree.

Value

A numeric vector

Examples

```
## calculate dg for inventory data grouped by plot_id and species
library(dplyr)
inventory_samples |>
mutate(dclass = silv_tree_dclass(diameter)) |>
  summarise(
    height = mean(height, na.rm = TRUE),
    ntrees = n(),
    .by    = c(plot_id, species, dclass)
  ) |>
  mutate(
    ntrees_ha = silv_density_ntrees_ha(ntrees, plot_size = 10),
    h0        = silv_stand_dominant_height(dclass, height, ntrees_ha),
    dg        = silv_stand_qmean_diameter(dclass, ntrees_ha),
    .by       = c(plot_id, species)
  )

## calculate dg for a vector of diameters
silv_stand_qmean_diameter(c(12.5, 23.5, 14, 16, 18.5))
```

`silv_summary` *Calculates a bunch of forest metrics*

Description

Summarize forest inventory data calculating most typical variables

Usage

```
silv_summary(
  data,
  diameter,
  height,
  plot_size,
  .groups = NULL,
  plot_shape = "circular",
  dmin = 7.5,
  dmax = NULL,
  class_length = 5,
  include_lowest = TRUE,
  which_h0 = "assman",
  which_spacing = "hart"
)
```

Arguments

`data` A tibble of inventory data

`diameter` A column with inventory diameters

`height` A column with inventory heights

`plot_size` The size of the plot. See `silv_density_ntrees_ha()`

`.groups` A character vector with variables to group by (e.g. plot id, tree species, etc)

`plot_shape` The shape of the sampling plot. Either `circular` or `rectangular`

`dmin` The minimum inventory diameter in centimeters

`dmax` The maximum inventory diameter in centimeters. Values that are greater than `dmax` are included in the greatest class

`class_length` The length of the class in centimeters

`include_lowest` Logical. If TRUE (the default), the intervals are `[dim1, dim2)`. If FALSE, the intervals are `(dim1, dim2]`

[dim1, dim2)`. If FALSE, the intervals are `(dim1, dim2]: R:dim1,%20dim2)%60.%20If%20FALSE

`which_h0` The method to calculate the dominant height. See `silv_stand_dominant_height()`

`which_spacing` A character with the name of the index (either `hart` or `hart-brecking`). See `silv_density_hart()`

Details

The function calculates many inventory parameters and returns two tibbles:

- **dclass_metrics**: metrics summarized by .groups and diametric classes
- **group_metrics**: metrics summarized by .groups

Value

an S7 `Inventory` list with 2 `tibbles`

Examples

```
silv_summary(
  data      = inventory_samples,
  diameter  = diameter,
  height    = height,
  plot_size = 10,
  .groups   = c("plot_id", "species")
 )
```

silv_treatment_thinning

Calculate Forestry Thinning Schemes

Description

Calculates thinning schemes for forest management by selecting trees to extract based on specified criteria. Supports both thinning from below (removing smaller trees) and thinning from above (removing larger trees) approaches.

Usage

```
silv_treatment_thinning(
  data,
  var,
  diameter,
  ntrees,
  thinning = "below",
  perc = 0.3,
  .groups = NULL
)
```

Arguments

`data`	A data frame, or silviculture::Inventory object. See details.
`var`	A variable used for calculating the thinning. Typically used variables basal area, number of trees, or volume
`diameter`	Numeric vector with diametric classes
`ntrees`	Numeric vector with the number of trees per hectare of each diametric class
`thinning`	Charater string specifying the thinning type. Available options are `below` and `above`
`perc`	Numeric value between 0 and 1 specifying the percentage of `var` to extract
`.groups`	A character vector with variables to group by (e.g. plot id, tree species, etc). Ignored when using a `silviculture::Inventory` object

Details

This function implements common silvicultural thinning practices:

Thinning from below: Removes trees with the lowest values of the specified variable. This approach typically removes suppressed, damaged, or poor-quality trees, mimicking natural mortality processes.

Thinning from above: Removes trees with the highest values of the specified variable. This approach harvests the most valuable trees while leaving smaller trees to continue growing.

The function calculates which trees to extract based on the ranking of the specified variable and the desired thinning percentage. When grouping variables are provided, thinning is calculated separately for each group.

Using a silviculture::Inventory object The result of `silv_summary()` can be used as the `data` argument. If so, the `.groups` will be taken from this object, and it will keep the previous data in a new S7 object.

Value

A `silviculture::Thinning` object with three items:

- **data**: the input data with two new columns
- **group_metrics**: it will include the data from the `silviculture::Inventory` object
- **thinning_opts**: options used for S7 methods

See Also

`silv_summary()`

Examples

```
# Get summary of inventory data
inventory <- inventory_samples |>
 silv_summary(
   diameter  = diameter,
   height    = height,
   plot_size = 25,
   .groups   = c('plot_id', 'species')
 )

## Thinning from below removing 30% of trees based on basal area
silv_treatment_thinning(
  data     = inventory,
  var      = g_ha,
  diameter = dclass,
  ntrees   = ntrees_ha,
  thinning = "below",
  perc     = 0.3
)

## Thinning from above removing 20% of trees based on basal area
silv_treatment_thinning(
  data     = inventory,
  var      = g_ha,
  diameter = dclass,
  ntrees   = ntrees_ha,
  thinning = "above",
  perc     = 0.2
)
```

`silv_tree_basal_area` *Calculates Basal Area*

Description

Calculates Basal Area in square meters.

Usage

```
silv_tree_basal_area(diameter, units = "cm")
```

Arguments

`diameter`	Numeric vector of diameters or diameter classes
`units`	The units of the diameter (one of `mm`, `cm`, `dm`, or `m`)

Details

The function uses the next formula:

$g = \frac{\pi}{40000} \cdot D^2$

where g is the basal area in m^2 of one tree, and D is the diameter in `cm`.

If you want to calculate the basal area for a group of trees (e.g. per hectares), please use `silv_stand_basal_area()`

Value

A numeric vector

See Also

`silv_stand_basal_area()`

Examples

```
## calculate individual basal area
silv_tree_basal_area(c(23, 11, 43.5, 94))
```

`silv_tree_dclass` *Classify diameters in classes*

Description

Classifies the measured diameters into classes of a specified length

Usage

```
silv_tree_dclass(
  diameter,
  dmin = 7.5,
  dmax = NULL,
  class_length = 5,
  include_lowest = TRUE,
  return_intervals = FALSE
)
```

Arguments

`diameter` A numeric vector of diameters

`dmin` The minimum inventory diameter in centimeters

`dmax` The maximum inventory diameter in centimeters. Values that are greater than `dmax` are included in the greatest class

`class_length` The length of the class in centimeters

`include_lowest` Logical. If TRUE (the default), the intervals are `[dim1, dim2)`. If FALSE, the intervals are `(dim1, dim2]`

[dim1, dim2)`. If FALSE, the intervals are `(dim1, dim2]: R:dim1,%20dim2)%60.%20If%20FALSE

`return_intervals`

If FALSE, it returns the intervals. Otherwise (the default), it returns the class center

Value

A numeric vector

Examples

```
library(dplyr)
inventory_samples |>
  mutate(dclass = silv_tree_dclass(diameter))
```

`silv_tree_volume` *Calculate Tree Volume*

Description

This function calculates the volume of a tree or logs using different formulas: Pressler, Huber, Smalian, and Newton. The appropriate diameter and height parameters must be provided depending on the selected formula.

Usage

```
silv_tree_volume(
  diameter_base = NULL,
  diameter_top = NULL,
  diameter_center = NULL,
  diameter = NULL,
  height = NULL,
  formula = "pressler",
  ntrees = NULL
)
```

Arguments

`diameter_base`	A numeric vector. The diameter at the base of the tree (required for Pressler, Smalian, and Newton formulas).
`diameter_top`	A numeric vector. The diameter at the top of the tree (required for Smalian and Newton formulas).
`diameter_center`	A numeric vector. The diameter at the center of the tree (required for Huber and Newton formulas).
`diameter`	A numeric vector. The diameter at breast height (used in Pressler formula if provided instead of `diameter_base`).
`height`	A numeric vector. The tree or log height (required for all formulas).
`formula`	Character. The volume formula to use. Options: `"pressler"`, `"huber"`, `"smalian"`, `"newton"`. Default is `"pressler"`.
`ntrees`	A numeric vector with number of trees of the same dimensions. Default is 1.

Value

A numeric value representing the tree volume.

Examples

```
silv_tree_volume(diameter_base = 30, height = 20, formula = "pressler")
silv_tree_volume(diameter_center = 25, height = 15, formula = "huber")
silv_tree_volume(diameter_base = 30, diameter_top = 20, height = 20, formula = "smalian")
```

`silv_volume` *Calculate Tree Volume*

Description

[Deprecated]

This function calculates the volume of a tree or logs using different formulas: Pressler, Huber, Smalian, and Newton. The appropriate diameter and height parameters must be provided depending on the selected formula.

Usage

```
silv_volume(
  diameter_base = NULL,
  diameter_top = NULL,
  diameter_center = NULL,
  diameter = NULL,
  height = NULL,
  formula = "pressler",
  ntrees = NULL
)
```

Arguments

`diameter_base` A numeric vector. The diameter at the base of the tree (required for Pressler, Smalian, and Newton formulas).

`diameter_top` A numeric vector. The diameter at the top of the tree (required for Smalian and Newton formulas).

`diameter_center` A numeric vector. The diameter at the center of the tree (required for Huber and Newton formulas).

`diameter` A numeric vector. The diameter at breast height (used in Pressler formula if provided instead of `diameter_base`).

`height` A numeric vector. The tree or log height (required for all formulas).

`formula` Character. The volume formula to use. Options: `"pressler"`, `"huber"`, `"smalian"`, `"newton"`. Default is `"pressler"`.

`ntrees` A numeric vector with number of trees of the same dimensions. Default is 1.

Value

A numeric value representing the tree volume.

Examples

```
silv_volume(diameter_base = 30, height = 20, formula = "pressler")
silv_volume(diameter_center = 25, height = 15, formula = "huber")
silv_volume(diameter_base = 30, diameter_top = 20, height = 20, formula = "smalian")
```

Index