Package ‘MLFS’

October 12, 2022

Type Package

Title Machine Learning Forest Simulator

Version 0.4.2

Author Jernej Jevsenak

Maintainer Jernej Jevsenak <jernej.jevsenak@gmail.com>

Description Climate-sensitive forest simulator based on the principles of machine learning. It stimulates all key processes in the forest: radial growth, height growth, mortality, crown recession, regeneration and harvesting. The method for predicting tree heights was described by Skudnik and Jevšenak (2022) <doi:10.1016/j.foreco.2022.120017>, while the method for predicting basal area increments (BAI) was described by Jevšenak and Skudnik (2021) <doi:10.1016/j.foreco.2020.118601>.

License GPL-3

Imports brnn(>= 0.6), ranger(>= 0.13.1), reshape2 (>= 1.4.4), pscl (>= 1.5.5), naivebayes(>= 0.9.7), magrittr(>= 0.7.0), dplyr(>= 1.0.0), tidyr (>= 1.1.3), tidyselect(>= 1.0.0)

Encoding UTF-8

LazyData true

Depends R(>= 3.4)

NeedsCompilation no

Repository CRAN

RoxygenNote 7.1.2

Date/Publication 2022-04-20 08:22:37 UTC

R topics documented:

add_stand_variables ............................................. 2
BAI_prediction .................................................. 3
calculate_BAL ...................................................... 5
crownHeight_prediction ....................................... 5
data_BAI .......................................................... 7
add_stand_variables

Description

This function adds two variables to existing data frame of individual tree measurements: 1) stand basal area and 2) the number of trees per hectare

Usage

add_stand_variables(df)

Arguments

df a data frame with individual tree measurements that include basal area and the upscale factors. All trees should also be described with plotID and year variables
Value

a data frame with added stand variables: total stand basal area and the number of trees per hectare

Examples

data(data_v1)
data_v1 <- add_stand_variables(df = data_v1)

Description

The Basal Area Increment BAI sub model that is run within the MLFS

Usage

```r
BAI_prediction(
  df_fit, df_predict,
  species_n_threshold = 100,
  site_vars,
  include_climate,
  eval_model_BAI = TRUE,
  rf_mtry = NULL,
  k = 10,
  blocked_cv = TRUE,
  measurement_thresholds = NULL,
  area_correction = NULL
)
```

Arguments

- `df_fit` a data frame with Basal Area Increments (BAI) and all independent variables as specified with the formula
- `df_predict` data frame which will be used for BAI predictions
- `species_n_threshold` a positive integer defining the minimum number of observations required to treat a species as an independent group
- `site_vars` a character vector of variable names which are used as site descriptors
- `include_climate` logical, should climate variables be included as predictors
- `eval_model_BAI` logical, should the the BAI model be evaluated and returned as the output
rf_mtry: a number of variables randomly sampled as candidates at each split of a random forest model for predicting basal area increments (BAI). If NULL, default settings are applied.

k: the number of folds to be used in the k-fold cross-validation.

blocked_cv: logical, should the blocked cross-validation be used in the evaluation phase?

measurement_thresholds: data frame with two variables: 1) DBH_threshold and 2) weight. This information is used to assign the correct weights in BAI and increment sub-model; and to upscale plot-level data to hectares.

area_correction: an optional data frame with three variables: 1) plotID and 2) DBH_threshold and 3) the correction factor to be multiplied by weight for this particular category.

Value

a list with four elements:

1. $predicted_BAI - a data frame with calculated basal area increments (BAI)
2. $eval_BAI - a data frame with predicted and observed basal area increments (BAI), or a character string indicating that BAI model was not evaluated
3. $rf_model_species - the output model for BAI (species level)
4. $rf_model_speciesGroups - the output model for BAI (species group level)

```r
# add BA to measurement thresholds
measurement_thresholds$BA_threshold <- ((measurement_thresholds$DBH_threshold/2)^2 * pi)/10000

BAI_outputs <- BAI_prediction(df_fit = data_BAI, df_predict = data_v6, site_vars = c("slope", "elevation", "northness", "siteIndex"), rf_mtry = 3, species_n_threshold = 100, include_climate = TRUE, eval_model_BAI = FALSE, k = 10, blocked_cv = TRUE, measurement_thresholds = measurement_thresholds)

# get the ranger objects
BAI_outputs_model_species <- BAI_outputs$rf_model_species BAI_outputs_model_groups <- BAI_outputs$rf_model_speciesGroups
```

Examples

```r
library(MLFS)
data(data_BAI)
data(data_v6)
data(measurement_thresholds)
```
**calculate_BAL**

**Description**

This function calculates the competition index BAL (Basal Area in Large trees) and adds it to the table of individual tree measurements that include basal area and the upscale factors. All trees should also be described with plotID and year variables.

**Usage**

```r
calculate_BAL(df)
```

**Arguments**

- `df`: a data frame with individual tree measurements that include basal area and the upscale factors. All trees should also be described with plotID and year variables.

**Value**

a data frame with calculated basal area in large trees (BAL)

**Examples**

```r
data(data_v1)
data_v1 <- calculate_BAL(df = data_v1)
```

---

**crownHeight_prediction**

**Description**

Model for predicting crown height

**Usage**

```r
crownHeight_prediction(
df_fit, 
df_predict, 
site_vars = site_vars, 
species_n_threshold = 100, 
k = 10, 
eval_model_crownHeight = TRUE, 
crownHeight_model = "lm",
```
crownHeight_prediction

BRNN_neurons = 3,
blocked_cv = TRUE
)

Arguments

df_fit data frame with tree heights and basal areas for individual trees
df_predict data frame which will be used for predictions
site_vars optional, character vector with names of site variables
species_n_threshold a positive integer defining the minimum number of observations required to treat
a species as an independent group
k the number of folds to be used in the k fold cross-validation
eval_model_crownHeight logical, should the crown height model be evaluated and returned as the output
crownHeight_model character string defining the model to be used for crown heights. Available are
ANN with Bayesian regularization (brnn) or linear regression (lm)
BRNN_neurons positive integer defining the number of neurons to be used in the brnn method.
blocked_cv logical, should the blocked cross-validation be used in the evaluation phase?

Value

a list with four elements:
1. $predicted_crownHeight - a data frame with imputed crown heights
2. $eval_crownHeight - a data frame with predicted and observed crown heights, or a character
   string indicating that crown height model was not evaluated
3. $model_species - the output model for crown heights (species level)
4. $model_speciesGroups - the output model for crown heights (species group level)

Examples

library(MLFS)
data(data_tree_heights)
data(data_v3)

# A) Example with linear model
Crown_h_predictions <- crownHeight_prediction(df_fit = data_tree_heights,
df_predict = data_v3,
crownHeight_model = "lm",
site_vars = c(),
species_n_threshold = 100,
k = 10, blocked_cv = TRUE,
eval_model_crownHeight = TRUE)
predicted_df <- Crown_h_predictions$predicted_crownHeight # df with imputed heights
evaluation_df <- Crown_h_predictions$eval_crownHeight # df with evaluation results
# B) Example with non-linear BRNN model

```r
crown_h_predictions <- crownHeight_prediction(df_fit = data_tree_heights, 
    df_predict = data_v3, 
    crownHeight_model = "brnn", 
    BRNN_neurons = 3, 
    site_vars = c(), 
    species_n_threshold = 100, 
    k = 10, blocked_cv = TRUE, 
    eval_model_crownHeight = TRUE)
```

---

**data_BAI**

An example of joined national forest inventory data, site descriptors, and climate data that is used as a fitting data frame for BAI sub model

---

**Description**

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for BAI sub model. To make examples running more quickly, we keep only one tree species: PINI.

**Usage**

`data_BAI`

**Format**

A data frame with 135 rows and 25 variables:

- **plotID** a unique identifier for plot
- **treeID** a unique identifier for tree
- **year** year in which plot was visited
- **speciesGroup** identifier for species group
- **code** status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)
- **species** species name
- **height** tree height in meters
- **crownHeight** crown height in meters
- **protected** logical, 1 if protected, otherwise 0
- **slope** slope on a plot
- **elevation** plot elevation
- **northness** plot northness, 1 is north, 0 is south
- **siteIndex** a proxy for site index, higher value represents more productive sites
- **BA** basal area of individual trees in m²
### data_climate

**Weight**  upscale weight to calculate hectare values

**Stand BA** Total stand basal area

**Stand n** The number of trees in a stand

**BAL** Basal Area in Large trees

**p_BA** basal area of individual trees in m2 from previous simulation step

**p_height** tree height in meters from previous simulation step

**p_crownHeight** crown height in meters from previous simulation step

**p_weight** upscale weight to calculate hectare values from previous simulation step

**BAI** basal area increment

**p_sum** monthly precipitation sum

**t_avg** monthly mean temperature

---

**data_climate**

*An example of climate data*

### Description

This is simulated monthly climate data, and consists of precipitation sum and mean temperature

### Usage

data_climate

### Format

A data frame with 16695 rows and 5 variables:

- **plotID** a unique identifier for plot
- **year** year
- **month** month
- **t_avg** monthly mean temperature
- **p_sum** monthly precipitation sum
An example of data_final_cut_weights

Description

Each species should have one weight that is multiplied with the probability of being harvested when final_cut is applied

Usage

data_final_cut_weights

Format

A data frame with 36 rows and 6 variables:

- species  species name as used in data_NFI
- step_1  final cut weight applied in step 1
- step_2  final cut weight applied in step 2
- step_3  final cut weight applied in step 3
- step_4  final cut weight applied in step 4
- step_5  final cut weight applied in step 5 and all subsequent steps

An example of data_ingrowth suitable for the MLFS

Description

An example of plot-level data with plotID, stand variables and site descriptors, and the two target variables describing the number of ingrowth trees for inner (ingrowth_3) and outer (ingrowth_15) circles

Usage

data_ingrowth
Format

A data frame with 365 rows and 11 variables:

- plotID: a unique identifier for plot
- year: year in which plot was visited
- stand_BA: Total stand basal area
- stand_n: The number of trees in a stand
- BAL: Basal Area in Large trees
- slope: slope on a plot
- elevation: plot elevation
- siteIndex: a proxy for site index, higher value represents more productive sites
- northness: plot northness, 1 is north, 0 is south
- ingrowth_3: the number of new trees in inner circle
- ingrowth_15: the number of new trees in outer circle

Description

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for mortality sub model

Usage

data_mortality

Format

A data frame with 6394 rows and 25 variables:

- plotID: a unique identifier for plot
- treeID: a unique identifier for tree
- year: year in which plot was visited
- speciesGroup: identifier for species group
- code: status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)
- species: species name
- height: tree height in meters
- crownHeight: crown height in meters
- protected: logical, 1 if protected, otherwise 0
data_NFI

slope  slope on a plot

elevation  plot elevation

northness  plot northness, 1 is north, 0 is south

siteIndex  a proxy for site index, higher value represents more productive sites

BA  basal area of individual trees in m2

weight  upscale weight to calculate hectare values

stand_BA  Total stand basal area

stand_n  The number of trees in a stand

BAL  Basal Area in Large trees

p_BA  basal area of individual trees in m2 from previous simulation step

p_height  tree height in meters from previous simulation step

p_crownHeight  crown height in meters from previous simulation step

p_weight  upscale weight to calculate hectare values from previous simulation step

BAI  basal area increment

p_sum  monthly precipitation sum

t_avg  monthly mean temperature

---

An example of national forest inventory data

Description

This is simulated data that reassemble the national forest inventory

Usage

data_NFI

Format

A data frame with 11984 rows and 10 variables:

plotID  a unique identifier for plot

treeID  a unique identifier for tree

year  year in which plot was visited

speciesGroup  identifier for species group

code  status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)

DBH  diameter at breast height in cm

species  species name

height  tree height in meters

crownHeight  crown height in meters

protected  logical, 1 if protected, otherwise 0
data_site

An example of site descriptors

Description
This is simulated data describing site descriptors

Usage
data_site

Format
A data frame with 371 rows and 5 variables:
- plotID a unique identifier for plot
- slope slope on a plot
- elevation plot elevation
- northness plot northness, 1 is north, 0 is south
- siteIndex a proxy for site index, higher value represents more productive sites

data_tariffs

An example of table with one-parametric volume functions (adapted uniform French tariffs)

Description
The adapted uniform French tariffs are typically used in Slovenia to determine tree volume based on tree DBH

Usage
data_tariffs

Format
A data frame with 1196 rows and 4 variables:
- tarifa_class tariff class for a particular species on this plot
- plotID plot identifier
- species species name as used in data_NFI
- v45 volume of tree with DBH 45 cm
**data_thinning_weights**  
*An example of data_thinning_weights*

**Description**

Each species should have one weight that is multiplied with the probability of being harvested when thinning is applied.

**Usage**

data_thinning_weights

**Format**

A data frame with 36 rows and 6 variables:

- **species**  species name as used in data_NFI
- **step_1**  thinning weight applied in step 1
- **step_2**  thinning weight applied in step 2
- **step_3**  thinning weight applied in step 3
- **step_4**  thinning weight applied in step 4
- **step_5**  thinning weight applied in step 5 and all subsequent steps

---

**data_tree_heights**  
*An example of data with individual tree and crown heights that can be used as a fitting data frame for predicting tree and crown heights in MLFS*

**Description**

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for some specific functions.

**Usage**

data_tree_heights
Format

A data frame with 2741 rows and 8 variables:

- **plotID** a unique identifier for plot
- **treeID** a unique identifier for tree
- **year** year in which plot was visited
- **speciesGroup** identifier for species group
- **species** species name
- **height** tree height in meters
- **crownHeight** crown height in meters
- **BA** basal area of individual trees in m²

---

**data_v1**

An example of joined national forest inventory and site data that is used within the MLFS

---

Description

This is simulated data that reassemble the national forest inventory and simulated data. We use it to show how to run examples for some specific functions

Usage

data_v1

Format

A data frame with 11984 rows and 15 variables:

- **plotID** a unique identifier for plot
- **treeID** a unique identifier for tree
- **year** year in which plot was visited
- **speciesGroup** identifier for species group
- **code** status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)
- **species** species name
- **height** tree height in meters
- **crownHeight** crown height in meters
- **protected** logical, 1 if protected, otherwise 0
- **slope** slope on a plot
- **elevation** plot elevation
- **northness** plot northness, 1 is north, 0 is south
- **siteIndex** a proxy for site index, higher value represents more productive sites
- **BA** basal area of individual trees in m²
- **weight** upscale weight to calculate hectare values
An example of joined national forest inventory and site data that is used within the MLFS

Description

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for tree and crown height predictions.

Usage

data_v2

Format

A data frame with 6948 rows and 14 variables:

- **plotID** a unique identifier for plot
- **treeID** a unique identifier for tree
- **year** year in which plot was visited
- **speciesGroup** identifier for species group
- **code** status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)
- **species** species name
- **height** tree height in meters
- **crownHeight** crown height in meters
- **BA** basal area of individual trees in m2
- **weight** upscale weight to calculate hectare values
- **p_BA** basal area of individual trees in m2 from previous simulation step
- **p_weight** upscale weight to calculate hectare values from previous simulation step
- **p_height** tree height in meters from previous simulation step
- **p_crownHeight** crown height in meters from previous simulation step
An example of joined national forest inventory and site data that is used within the MLFS

Description

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for tree and crown height predictions. The difference between data_v2 and data_v3 is that in data_v3, tree heights are already predicted.

Usage

data_v3

Format

A data frame with 6948 rows and 14 variables:

- **plotID** a unique identifier for plot
- **treeID** a unique identifier for tree
- **year** year in which plot was visited
- **speciesGroup** identifier for species group
- **code** status of a tree: 0 (normal), 1 (harvested), 2 (dead), 3 (ingrowth)
- **species** species name
- **height** tree height in meters
- **crownHeight** crown height in meters
- **BA** basal area of individual trees in m²
- **weight** upscale weight to calculate hectare values
- **p_BA** basal area of individual trees in m² from previous simulation step
- **p_height** tree height in meters from previous simulation step
- **p_crownHeight** crown height in meters from previous simulation step
- **p_weight** upscale weight to calculate hectare values from previous simulation step
- **volume** tree volume in m³
- **p_volume** tree volume in m³ from previous simulation step
**Description**

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for predicting tree mortality. Mortality occurs in the middle of a simulation step, so all variables have the preposition 'mid'.

**Usage**

`data_v4`

**Format**

A data frame with 6855 rows and 41 variables:

- **year**: year in which plot was visited
- **plotID**: a unique identifier for plot
- **treeID**: a unique identifier for tree
- **speciesGroup**: identifier for species group
- **code**: status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)
- **species**: species name
- **slope**: slope on a plot
- **elevation**: plot elevation
- **northness**: plot northness, 1 is north, 0 is south
- **siteIndex**: a proxy for site index, higher value represents more productive sites
- **p_sum**: monthly precipitation sum
- **t_avg**: monthly mean temperature
- **BA_mid**: basal area of individual trees in m2 in the middle of a simulation step
- **BAI_mid**: basal area increment in the middle of a simulation step
- **weight_mid**: upscale weight to calculate hectare values in the middle of a simulation step
- **height_mid**: tree height in meters in the middle of a simulation step
- **crownHeight_mid**: crown height in meters in the middle of a simulation step
- **volume_mid**: tree volume in m3 in the middle of a simulation step
- **BAL_mid**: Basal Area in Large trees the middle of a simulation step
- **stand_BA_mid**: Total stand basal area the middle of a simulation step
- **stand_n_mid**: The number of trees in a stand the middle of a simulation step
### data_v5

An example of joined national forest inventory and site data that is used within the MLFS

#### Description

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for simulating harvesting.

#### Usage

```r
data_v5
```

#### Format

A data frame with 5949 rows and 10 variables:

- `species` species name
- `year` year in which plot was visited
- `plotID` a unique identifier for plot
- `treeID` a unique identifier for tree
- `speciesGroup` identifier for species group
- `code` status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)
- `volume_mid` tree volume in m³ in the middle of a simulation step
- `weight_mid` upscale weight to calculate hectare values in the middle of a simulation step
- `BA_mid` basal area of individual trees in m² in the middle of a simulation step
- `protected` logical, 1 if protected, otherwise 0

### data_v6

An example of joined national forest inventory and site data that is used within the MLFS

#### Description

This is simulated data that reassemble the national forest inventory data. We use it to show how to run examples for simulating Basal Area Increments (BAI) and the ingrowth of new trees. To make examples running more quickly, we keep only one tree species: PINI

#### Usage

```r
data_v6
```
A data frame with 186 rows and 27 variables:

- **species**: species name
- **year**: year in which plot was visited
- **plotID**: a unique identifier for plot
- **treeID**: a unique identifier for tree
- **speciesGroup**: identifier for species group
- **code**: status of a tree: 0 (normal), 1(harvested), 2(dead), 3 (ingrowth)
- **height**: tree height in meters
- **crownHeight**: crown height in meters
- **protected**: logical, 1 if protected, otherwise 0
- **slope**: slope on a plot
- **elevation**: plot elevation
- **northness**: plot northness, 1 is north, 0 is south
- **siteIndex**: a proxy for site index, higher value represents more productive sites
- **BA**: basal area of individual trees in m2
- **weight**: upscale weight to calculate hectare values
- **stand_BA**: Total stand basal area
- **stand_n**: The number of trees in a stand
- **BAL**: Basal Area in Large trees
- **p_BA**: basal area of individual trees in m2 from previous simulation step
- **p_height**: tree height in meters from previous simulation step
- **p_crownHeight**: crown height in meters from previous simulation step
- **p_weight**: upscale weight to calculate hectare values from previous simulation step
- **BAI**: basal area increment
- **p_sum**: monthly precipitation sum
- **t_avg**: monthly mean temperature
- **volume**: tree volume in m3
- **p_volume**: tree volume in m3 from previous simulation step
An example table with parameters and equations for n-parametric volume functions

Description

Volume functions can be specified for each species and plot separately, also limited to specific DBH interval. The factor variables (vol_factor, h_factor and DBH_factor) are used to control the input and output units.

Usage

df_volume_parameters

Format

A data frame with 6 rows and 14 variables:

- **species**: species name as used in data_NFI. The category REST is used for all species without specific equation
- **equation**: equation for selected volume function
- **vol_factor**: will be multiplied with the volume
- **h_factor**: will be multiplied with tree height
- **d_factor**: will be divided with tree DBH
- **DBH_min**: lower interval threshold for considered trees
- **DBH_max**: upper interval threshold for considered trees
- **a**: parameter a for volume equation
- **b**: parameter b for volume equation
- **c**: parameter c for volume equation
- **d**: parameter d for volume equation
- **e**: parameter e for volume equation
- **f**: parameter f for volume equation
- **g**: parameter g for volume equation
An example table with form factors used to calculate tree volume

**Description**

Form factors can be specified per species, plot or per species and plot

**Usage**

```r
form_factors
```

**Format**

A data frame with 1199 rows and 3 variables:

- `plotID`  a unique identifier for plot
- `species` species name as used in data_NFI
- `form` for factor used to calculate tree volume

**Description**

Height model

**Usage**

```r
height_prediction(
  df_fit,
  df_predict,
  species_n_threshold = 100,
  height_model = "naslund",
  BRNN_neurons = 3,
  height_pred_level = 0,
  eval_model_height = TRUE,
  blocked_cv = TRUE,
  k = 10
)
```
height_prediction

Arguments

- **df_fit** data frame with tree heights and basal areas for individual trees
- **df_predict** data frame which will be used for predictions
- **species_n_threshold** a positive integer defining the minimum number of observations required to treat a species as an independent group
- **height_model** character string defining the model to be used for height prediction. If 'brnn', then ANN method with Bayesian Regularization is applied. In addition, all 2- and 3-parametric H-D models from lmfor R package are available.
- **BRNN_neurons** positive integer defining the number of neurons to be used in the brnn method.
- **height_pred_level** integer with value 0 or 1 defining the level of prediction for height-diameter (H-D) models. The value 1 defines a plot-level prediction, while the value 0 defines regional-level predictions. Default is 0. If using 1, make sure to have representative plot-level data for each species.
- **eval_model_height** logical, should the height model be evaluated and returned as the output
- **blocked_cv** logical, should the blocked cross-validation be used in the evaluation phase?
- **k** the number of folds to be used in the k fold cross-validation

Value

a list with four elements:

1. $data_height_predictions - a data frame with imputed tree heights
2. $data_height_eval - a data frame with predicted and observed tree heights, or a character string indicating that tree heights were not evaluated
3. $model_species - the output model for tree heights (species level)
4. $model_speciesGroups - the output model for tree heights (species group level)

Examples

```r
library(MLFS)
data(data_tree_heights)
data(data_v2)

# A) Example with the BRNN method
h_predictions <- height_prediction(df_fit = data_tree_heights,
df_predict = data_v2,
species_n_threshold = 100,
height_pred_level = 0,
height_model = "brnn",
BRNN_neurons = 3,
eval_model_height = FALSE,
blocked_cv = TRUE, k = 10 )
```

ingrowth_parameter_list

An example data of `ingrowth_parameter_list`

Description

This is a list with two ingrowth levels: 3 (inner circle) and 15 (outer circle). In each list there are deciles of DBH distributions that are used to simulate DBH for new trees, separately for each ingrowth category.

Usage

`ingrowth_parameter_list`

Format

A list with 2 elements:

3  deciles of DBH distribution for ingrowth category 3
15  deciles of DBH distribution for ingrowth category 15

ingrowth_table

An example data of `ingrowth_table`

Description

Ingrowth table is used within the ingrowth sub model to correctly simulate different ingrowth levels and associated upscale weights.

Usage

`ingrowth_table`

Format

A data frame with 2 rows and 4 variables:

code  ingrowth codes
DBH_threshold  a DBH threshold for particular ingrowth category
DBH_max  maximum DBH for a particular ingrowth category
weight  the upscale weight for particular measurement category
**max_size_data**

An example of data with maximum allowed BA that is used in the mortality sub model

**Description**

This is simulated max_size_data and used for examples in mortality sub model

**Usage**

`max_size_data`

**Format**

A data frame with 36 rows and 2 variables:

- **species** species name
- **BA_max** The maximum allowed basal area (BA) for each individual species

---

**measurement_thresholds**

An example of measurement_thresholds table

**Description**

An example of measurement_thresholds table resulting from concentric plots as used in Slovenian NFI

**Usage**

`measurement_thresholds`

**Format**

A data frame with 2 rows and 2 variables:

- **DBH_threshold** a DBH threshold for particular measurement category
- **weight** the upscale weight for particular measurement category
**Description**

Machine Learning Forest Simulator

**Usage**

```R
MLFS(
  data_NFI,
  data_site,
  data_tariffs = NULL,
  data_climate = NULL,
  df_volumeF_parameters = NULL,
  thinning_weights_species = NULL,
  final_cut_weights_species = NULL,
  thinning_weights_plot = NULL,
  final_cut_weights_plot = NULL,
  form_factors = NULL,
  form_factors_level = "species_plot",
  uniform_form_factor = 0.42,
  sim_steps,
  volume_calculation = "volume_functions",
  merchantable_whole_tree = "merchantable",
  sim_harvesting = TRUE,
  sim_mortality = TRUE,
  sim_ingrowth = TRUE,
  sim_crownHeight = TRUE,
  harvesting_sum = NULL,
  forest_area_ha = NULL,
  harvest_sum_level = NULL,
  plot_upscale_type = NULL,
  plot_upscale_factor = NULL,
  mortality_share = NA,
  mortality_share_type = "volume",
  mortality_model = "glm",
  ingrowth_model = "ZIF_poiss",
  BAI_rf_mtry = NULL,
  ingrowth_rf_mtry = NULL,
  mortality_rf_mtry = NULL,
  nb_laplace = 0,
  harvesting_type = "final_cut",
  share_thinning = 0.8,
  final_cut_weight = 10,
  thinning_small_weight = 1,
  species_n_threshold = 100,
)```
height_model = "brnn",
crownHeight_model = "brnn",
BRNN_neurons_crownHeight = 1,
BRNN_neurons_height = 3,
height_pred_level = 0,
include_climate = FALSE,
select_months_climate = c(1, 12),
set_eval_mortality = TRUE,
set_eval_crownHeight = TRUE,
set_eval_height = TRUE,
set_eval_ingrowth = TRUE,
set_eval_BAI = TRUE,
k = 10,
blocked_cv = TRUE,
max_size = NULL,
max_size_increase_factor = 1,
ingrowth_codes = c(3),
ingrowth_max_DBH_percentile = 0.9,
measurement_thresholds = NULL,
area_correction = NULL,
export_csv = FALSE,
sim_export_mode = TRUE,
include_mortality_BAI = TRUE,
intermediate_print = FALSE
)

Arguments

data_NFI data frame with individual tree variables
data_site data frame with site descriptors. This data is related to data_NFI based on the 'plotID' column
data_tariffs optional, but mandatory if volume is calculated using the one-parametric tariff functions. Data frame with plotID, species and V45. See details.
data_climate data frame with climate data, covering the initial calibration period and all the years which will be included in the simulation
df_volumeF_parameters optional, data frame with species-specific volume function parameters
thinning_weights_species data frame with thinning weights for each species. The first column represents species code, each next column consists of species-specific thinning weights applied in each simulation step
final_cut_weights_species data frame with final cut weights for each species. The first column represents species code, each next column consists of species-specific final cut weights applied in each simulation step
thinning_weights_plot data frame with harvesting weights related to plot IDs, used for thinning
final_cut_weights_plot
data frame with harvesting weights related to plot IDs, used for final cut
form_factors   optional, data frame with species-specific form factors
form_factors_level
character, the level of specified form factors. It can be 'species', 'plot' or 'species_plot'
uniform_form_factor
numeric, uniform form factor to be used for all species and plots. Only if form_factors are not provided
sim_steps   The number of simulation steps
volume_calculation
character string defining the method for volume calculation: 'tariffs', 'volume_functions', 'form_factors' or 'slo_2p_volume_functions'
merchantable_whole_tree
character, 'merchantable' or 'whole_tree'. It indicates which type of volume functions will be used. This parameter is used only for volume calculation using the 'slo_2p_volume_functions'.
sim_harvesting logical, should harvesting be simulated?
sim_mortality logical, should mortality be simulated?
sim_ingrowth logical, should ingrowth be simulated?
sim_crownHeight logical, should crown heights be simulated? If TRUE, a crownHeight column is expected in data_NFI
harvesting_sum a value, or a vector of values defining the harvesting sums through the simulation stage. If a single value, then it is used in all simulation steps. If a vector of values, the first value is used in the first step, the second in the second step, etc.
forest_area_ha   the total area of all forest which are subject of the simulation
harvest_sum_level
integer with value 0 or 1 defining the level of specified harvesting sum: 0 for plot level and 1 for regional level
plot_upscale_type
character defining the upscale method of plot level values. It can be 'area' or 'upscale factor'. If 'area', provide the forest area represented by all plots in hectares (forest_area_ha argument). If 'factor', provide the fixed factor to upscale the area of all plots. Please note: forest_area_ha/plot_upscale_factor = number of unique plots. This argument is important when harvesting sum is defined on regional level.
plot_upscale_factor numeric value to be used to upscale area of each plot
mortality_share a value, or a vector of values defining the proportion of the volume which is to be the subject of mortality. If a single value, then it is used in all simulation steps. If a vector of values, the first value is used in the first step, the second in the second step, and so on.
mortality_share_type
character, it can be 'volume' or 'n_trees'. If 'volume' then the mortality share relates to total standing volume, if 'n_trees' then mortality share relates to the total number of standing trees

mortality_model
model to be used for mortality prediction: 'glm' for generalized linear models; 'rf' for random forest algorithm; 'naiveBayes' for Naive Bayes algorithm

ingrowth_model
model to be used for ingrowth predictions. 'glm' for generalized linear models (Poisson regression), 'ZIF_poiss' for zero inflated Poisson regression and 'rf' for random forest

BAI_rf_mtry
a number of variables randomly sampled as candidates at each split of a random forest model for predicting basal area increments (BAI). If NULL, default settings are applied.

ingrowth_rf_mtry
a number of variables randomly sampled as candidates at each split of a random forest model for predicting ingrowth. If NULL, default settings are applied

mortality_rf_mtry
a number of variables randomly sampled as candidates at each split of a random forest model for predicting mortality. If NULL, default settings are applied

nb_laplace
value used for Laplace smoothing (additive smoothing) in naive Bayes algorithm. Defaults to 0 (no Laplace smoothing)

harvesting_type
character, it could be 'random', 'final_cut', 'thinning' or 'combined'. The latter combines 'final_cut' and 'thinning' options, where the share of each is specified with the argument 'share_thinning'

share_thinning
numeric, a number or a vector of numbers between 0 and 1 that specifies the share of thinning in comparison to final_cut. Only used if harvesting_type is 'combined'

final_cut_weight
numeric value affecting the probability distribution of harvested trees. Greater value increases the share of harvested trees having larger DBH. Default is 10.

thinning_small_weight
numeric value affecting the probability distribution of harvested trees. Greater value increases the share of harvested trees having smaller DBH. Default is 1.

species_n_threshold
a positive integer defining the minimum number of observations required to treat a species as an independent group

height_model
character string defining the model to be used for height prediction. If brnn, then ANN method with Bayesian Regularization is applied.

crownHeight_model
character string defining the model to be used for crown heights. Available are ANN with Bayesian regularization (brnn) or linear regression (lm)

BRNN_neurons_crownHeight
a positive integer defining the number of neurons to be used in the brnn method for predicting crown heights
**MLFS**

**BRNN_neurons_height**
a positive integer defining the number of neurons to be used in the brnn method for predicting tree heights

**height_pred_level**
integer with value 0 or 1 defining the level of prediction for height-diameter (H-D) models. The value 1 defines a plot-level prediction, while the value 0 defines regional-level predictions. Default is 0. If using 1, make sure to have representative plot-level data for each species.

**include_climate**
logical, should climate variables be included as predictors

**select_months_climate**
vector of subset months to be considered. Default is c(1,12), which uses all months.

**set_eval_mortality**
logical, should the mortality model be evaluated and returned as the output

**set_eval_crownHeight**
logical, should the crownHeight model be evaluated and returned as the output

**set_eval_height**
logical, should the height model be evaluated and returned as the output

**set_eval_ingrowth**
logical, should the ingrowth model be evaluated and returned as the output

**set_eval_BAI**
logical, should the BAI model be evaluated and returned as the output

**k**
the number of folds to be used in the k fold cross-validation

**blocked_cv**
logical, should the blocked cross-validation be used in the evaluation phase?

**max_size**
a data frame with the maximum values of DBH for each species. If a tree exceeds this value, it dies. If not provided, the maximum is estimated from the input data. Two columns must be present, i.e. 'species' and 'DBH_max'

**max_size_increase_factor**
numeric value, which will be used to increase the max DBH for each species, when the maximum is estimated from the input data. If the argument 'max_size' is provided, the 'max_size_increase_factor' is ignored. Default is 1. To increase maximum for 10 percent, use 1.1.

**ingrowth_codes**
numeric value or a vector of codes which refer to ingrowth trees

**ingrowth_max_DBH_percentile**
which percentile should be used to estimate the maximum simulated value of ingrowth trees?

**measurement_thresholds**
data frame with two variables: 1) DBH_threshold and 2) weight. This information is used to assign the correct weights in BAI and increment sub-model; and to upscale plot-level data to hectares.

**area_correction**
optional data frame with three variables: 1) plotID and 2) DBH_threshold and 3) the correction factor to be multiplied by weight for this particular category.

**export_csv**
logical, if TRUE, at each simulation step, the results are saved in the current working directory as csv file
sim_export_mode
logical, if FALSE, the results of the individual simulation steps are not merged into the final export table. Therefore, output element 1 ($sim_results) will be empty. This was introduced to allow simulations when using larger data sets and long term simulations that might exceed the available RAM. In such cases, we recommend setting the argument export_csv = TRUE, which will export each simulation step to the current working directory.

include_mortality_BAI
logical, should basal area increments (BAI) be used as independent variable for predicting individual tree morality?

intermediate_print
logical, if TRUE intermediate steps will be printed while MLFS is running

Value
a list of class mlfs with at least 15 elements:

1. $sim_results - a data frame with the simulation results
2. $height_eval - a data frame with predicted and observed tree heights, or a character string indicating that tree heights were not evaluated
3. $crownHeight_eval - a data frame with predicted and observed crown heights, or character string indicating that crown heights were not evaluated
4. $mortality_eval - a data frame with predicted and observed probabilities of dying for all individual trees, or character string indicating that mortality sub-model was not evaluated
5. $ingrowth_eval - a data frame with predicted and observed number of new ingrowth trees, separately for each ingrowth level, or character string indicating that ingrowth model was not evaluated
6. $BAI_eval - a data frame with predicted and observed basal area increments (BAI), or character string indicating that BAI model was not evaluated
7. $height_model_species - the output model for tree heights (species level)
8. $height_model_speciesGroups - the output model for tree heights (species group level)
9. $crownHeight_model_species - the output model for crown heights (species level)
10. $crownHeight_model_speciesGroups - the output model for crown heights (species group level)
11. $mortality_model - the output model for mortality
12. $BAI_model_species - the output model for basal area increments (species level)
13. $BAI_model_speciesGroups - the output model for basal area increments (species group level)
14. $max_size - a data frame with maximum allowed diameter at breast height (DBH) for each species
15. $ingrowth_model_3 - the output model for ingrowth (level 1) – the output name depends on ingrowth codes
16. $ingrowth_model_15 - the output model for ingrowth (level 2) – optional and the output name depends on ingrowth codes
Examples

```r
library(MLFS)

# open example data
data(data_NFI)
data(data_site)
data(data_climate)
data(data_volume_parameters)
data(measurement_thresholds)

test_simulation <- MLFS(data_NFI = data_NFI,
data_site = data_site,
data_climate = data_climate,
df_volumeF_parameters = df_volume_parameters,
form_factors = volume_functions,
sim_steps = 2,
sim_harvesting = TRUE,
harvesting_sum = 100000,
harvest_sum_level = 1,
plot_upscale_type = "factor",
plot_upscale_factor = 1600,
measurement_thresholds = measurement_thresholds,
ingrowth_codes = c(3,15),
volume_calculation = "volume_functions",
select_months_climate = seq(6,8),
intermediate_print = FALSE)
```

Description

ingrowth model for predicting new trees within the MLFS

Usage

```r
predict_ingrowth(
  df_fit,
  df_predict,
  site_vars = site_vars,
  include_climate = include_climate,
  eval_model_ingrowth = TRUE,
  k = 10,
  blocked_cv = TRUE,
  ingrowth_model = "glm",
  rf_mtry = NULL,
)```
predict_ingrowth

    ingrowth_table = NULL,
    DBH_distribution_parameters = NULL
)

Arguments

df_fit  a plot-level data with plotID, stand variables and site descriptors, and the two
target variables describing the number of ingrowth trees for inner (ingrowth_3)
and outer (ingrowth_15) circles
df_predict data frame which will be used for ingrowth predictions
site_vars a character vector of variable names which are used as site descriptors
include_climate logical, should climate variables be included as predictors
eval_model_ingrowth logical, should the ingrowth model be evaluated and returned as the output
k the number of folds to be used in the k fold cross-validation
blocked_cv logical, should the blocked cross-validation be used in the evaluation phase?
ingrowth_model model to be used for ingrowth predictions. 'glm' for generalized linear models
(Poisson regression), 'ZIF_poiss' for zero inflated Poisson regression and 'rf' for random forest
rf_mtry a number of variables randomly sampled as candidates at each split of a random
forest model for predicting ingrowth. If NULL, default settings are applied.
ingrowth_table a data frame with 4 variables: (ingrowth) code, DBH_threshold, DBH_max and
weight. Ingrowth table is used within the ingrowth sub model to correctly simulate different ingrowth levels and associated upscale weights
DBH_distribution_parameters A list with deciles of DBH distributions that are used to simulate DBH for new
trees, separately for each ingrowth category

Value

a list with four elements:

1. $predicted_ingrowth - a data frame with newly added trees based on the ingrowth predictions
2. $eval_ingrowth - a data frame with predicted and observed number of new trees, separately
   for each ingrowth level, or character string indicating that ingrowth model was not evaluated
3. $mod_ing_3 - the output model for predicting the ingrowth of trees with code 3
4. $mod_ing_15 - the output model for predicting the ingrowth of trees with code 15 (the output
   name depends on the code used for this particular ingrowth level)

Examples

library(MLFS)
data(data_v6)
predict_mortality

```r
ingrowth_outputs <- predict_ingrowth(
  df_fit = data_ingrowth,
  df_predict = data_v6,
  site_vars = c("slope", "elevation", "northness", "siteIndex"),
  include_climate = TRUE,
  eval_model_ingrowth = FALSE,
  rf_mtry = 3,
  k = 10, blocked_cv = TRUE,
  ingrowth_model = 'rf',
  ingrowth_table = ingrowth_table,
  DBH_distribution_parameters = ingrowth_parameter_list)
```

---

**Description**

This sub model first fits a binary model to derive the effects of individual tree, site and climate variables on mortality; and afterwards predict the probability of dying for each tree from `df_predict`.

**Usage**

```r
predict_mortality(
  df_fit,
  df_predict,
  df_climate,
  mortality_share = NA,
  mortality_share_type = "volume",
  include_climate,
  site_vars,
  select_months_climate = c(6, 8),
  mortality_model = "rf",
  nb_laplace = 0,
  sim_crownHeight = FALSE,
  k = 10,
  eval_model_mortality = TRUE,
  blocked_cv = TRUE,
  sim_mortality = TRUE,
  sim_step_years = 5,
  rf_mtry = NULL,
  df_max_size = NULL,
  ingrowth_codes = 3,
  include_mortality_BAI = TRUE,
)```
predict_mortality

intermediate_print = FALSE
)

Arguments

df_fit a data frame with individual tree data and site descriptors where code is used to specify a status of each tree
df_predict data frame which will be used for mortality predictions
df_climate data frame with monthly climate data
mortality_share a value defining the proportion of the volume which is to be the subject of mortality
mortality_share_type character, it can be ‘volume’ or ‘n_trees’. If ‘volume’ then the mortality share relates to total standing volume, if ‘n_trees’ then mortality share relates to the total number of standing trees
include_climate logical, should climate variables be included as predictors
site_vars a character vector of variable names which are used as site descriptors
select_months_climate vector of subset months to be considered. Default is c(1,12), which uses all months.
mortality_model logical, should the mortality model be evaluated and returned as the output
nb_laplace value used for Laplace smoothing (additive smoothing) in naive Bayes algorithm. Defaults to 0 (no Laplace smoothing).
sim_crownHeight logical, should crown heights be considered as a predictor variable? If TRUE, a crownHeight column is expected in data_NFI
k the number of folds to be used in the k fold cross-validation
eval_model_mortality logical, should the mortality model be evaluated and returned as the output
blocked_cv logical, should the blocked cross-validation be used in the evaluation phase?
sim_mortality logical, should mortality be simulated?
sim_step_years the simulation step in years
rf_mtry number of variables randomly sampled as candidates at each split of a random forest model. If NULL, default settings are applied.
df_max_size a data frame with the maximum BA values for each species. If a tree exceeds this value, it dies.
ingrowth_codes numeric value or a vector of codes which refer to ingrowth trees
include_mortality_BAI logical, should basal area increments (BAI) be used as independent variable for predicting individual tree morality?
intermediate_print logical, if TRUE intermediate steps will be printed while the mortality sub model is running
Value

a list with three elements:

1. $predicted_mortality - a data frame with updated tree status (code) based on the predicted mortality
2. $eval_mortality - a data frame with predicted and observed probabilities of dying for all individual trees, or character string indicating that mortality sub-model was not evaluated
3. $model_output - the output model for mortality

Examples

data("data_v4")
data("data_mortality")
data("max_size_data")
mortality_outputs <- predict_mortality(
df_fit = data_mortality,
df_predict = data_v4,
mortality_share_type = 'volume',
df_climate = data_climate,
site_vars = c("slope", "elevation", "northness", "siteIndex"),
sim_mortality = TRUE,
mortality_model = 'naiveBayes',
 nb_laplace = 0,
sim_crownHeight = TRUE,
mortality_share = 0.02,
include_climate = TRUE,
 select_months_climate = c(6,7,8),
 eval_model_mortality = TRUE,
k = 10, blocked_cv = TRUE,
sim_step_years = 6,
df_max_size = max_size_data,
ingrowth_codes = c(3,15),
include_mortality_BAI = TRUE)

df_predicted <- mortality_outputs$predicted_mortality
df_evaluation <- mortality_outputs$eval_mortality

# confusion matrix
table(df_evaluation$mortality, round(df_evaluation$mortality_pred, 0))

simulate_harvesting  A sub model to simulate harvesting within the MLFS

Description

Harvesting is based on probability sampling, which depends on the selected parameters and the size of a tree. Bigger trees have higher probability of being harvested when final cut is applied, while smaller trees have higher probability of being sampled in the case of thinning.
simulate_harvesting(
  df,
  harvesting_sum,
  df_thinning_weights_species = NULL,
  df_final_cut_weights_species = NULL,
  df_thinning_weights_plot = NULL,
  df_final_cut_weights_plot = NULL,
  harvesting_type = "random",
  share_thinning = 0.8,
  final_cut_weight = 1e+07,
  thinning_small_weight = 1e+05,
  harvest_sum_level = 1,
  plot_upscale_type,
  plot_upscale_factor,
  forest_area_ha
)

Arguments

- **df**: a data frame with individual tree data, which include basal areas in the middle of a simulation step, species name and code
- **harvesting_sum**: a value, or a vector of values defining the harvesting sums through the simulation stage. If a single value, then it is used in all simulation steps. If a vector of values, the first value is used in the first step, the second in the second step, etc.
- **df_thinning_weights_species**: data frame with thinning weights for each species. The first column represents species code, each next column consists of species-specific thinning weights
- **df_final_cut_weights_species**: data frame with final cut weights for each species. The first column represents species code, each next column consists of species-specific final cut weights
- **df_thinning_weights_plot**: data frame with harvesting weights related to plot IDs, used for thinning
- **df_final_cut_weights_plot**: data frame with harvesting weights related to plot IDs, used for final cut
- **harvesting_type**: character, it could be 'random', 'final_cut', 'thinning' or 'combined'. The latter combines 'final_cut' and 'thinning' options, where the share of each is specified with the argument 'share_thinning'
- **share_thinning**: numeric, a number between 0 and 1 that specifies the share of thinning in comparison to final_cut. Only used if harvesting_type is 'combined'
- **final_cut_weight**: numeric value affecting the probability distribution of harvested trees. Greater value increases the share of harvested trees having larger DBH. Default is 10.
- **thinning_small_weight**: numeric value affecting the probability distribution of harvested trees. Greater value increases the share of harvested trees having smaller DBH. Default is 1.
harvest_sum_level
   integer with value 0 or 1 defining the level of specified harvesting sum: 0 for plot level and 1 for regional level

plot_upscale_type
   character defining the upscale method of plot level values. It can be 'area' or 'upscale factor'. If 'area', provide the forest area represented by all plots in hectares (forest_area_ha argument). If 'factor', provide the fixed factor to upscale the area of all plots. Please note: forest_area_ha/plot_upscale_factor = number of unique plots. This argument is important when harvesting sum is defined on regional level.

plot_upscale_factor
   numeric value to be used to upscale area of each plot

forest_area_ha
   the total area of all forest which are subject of the simulation

Value
   a data frame with updated status (code) of all individual trees based on the simulation of harvesting

Examples

library(MLFS)
data(data_v5)

data_v5 <- simulate_harvesting(df = data_v5,
   harvesting_sum = 5500000,
   harvesting_type = "combined",
   share_thinning = 0.50,
   harvest_sum_level = 1,
   plot_upscale_type = "factor",
   plot_upscale_factor = 1600,
   final_cut_weight = 5,
   thinning_small_weight = 1)

Description
   The calculation of individual tree volume using form factors, which can be defined per species, per plot, or per species and per plot

Usage

volume_form_factors(
   df,
   form_factors = NULL,
volume_functions

form_factors_level = "species",
uniform_form_factor = 0.42
)

Arguments

  df        data frame with tree heights and basal areas for individual trees
  form_factors   data frame with for factors for species, plot or both
  form_factors_level
                character, the level of specified form factors. It can be 'species', 'plot' or 'species_plot'
  uniform_form_factor
                a uniform form factor to be applied to all trees. If specified, it overwrites the argument 'form_factors'

Value

  a data frame with calculated volume for all trees

Examples

library(MLFS)
data(data_v3)
data(form_factors)
data_v3 <- volume_form_factors(df = data_v3, form_factors = form_factors,
  form_factors_level = "species_plot")
summary(data_v3)

volume_functions

Description

  The calculation of individual tree volume using the n-parameter volume functions for the MLFS

Usage

  volume_functions(df, df_volumeF_parameters = NULL)

Arguments

  df                data frame with tree heights and basal areas for individual trees
  df_volumeF_parameters
                  data frame with equations and parameters for n-parametric volume functions
Value

a data frame with calculated volume for all trees

Examples

```r
library(MLFS)
data(data_v3)
data(df_volume_parameters)

data_v3 <- volume_functions(df = data_v3,
    df_volumeF_parameters = df_volume_parameters)
```

Description

One-parameter volume functions (tariffs) for the MLFS.

Usage

`volume_tariffs(df, data_tariffs)`

Arguments

- `df` : data frame with tree heights and basal areas for individual trees
- `data_tariffs` : data frame with plot- and species-specific parameters for the calculations of tree volume

Value

a data frame with calculated volume for all trees

Examples

```r
data(data_v3)
data(data_tariffs)
data_v3 <- volume_tariffs(df = data_v3, data_tariffs = data_tariffs)
```
Index

* datasets
  data_BAI, 7
  data_climate, 8
  data_final_cut_weights, 9
  data_ingrowth, 9
  data_mortality, 10
  data_NFI, 11
  data_site, 12
  data_tariffs, 12
  data_thinning_weights, 13
  data_tree_heights, 13
  data_v1, 14
  data_v2, 15
  data_v3, 16
  data_v4, 17
  data_v5, 18
  data_v6, 18
  df_volume_parameters, 20
  form_factors, 21
  ingrowth_parameter_list, 23
  ingrowth_table, 23
  max_size_data, 24
  measurement_thresholds, 24
  MLFS, 25
  predict_ingrowth, 31
  predict_mortality, 33
  simulate_harvesting, 35
  volume_form_factors, 37
  volume_functions, 38
  volume_tariffs, 39

add_stand_variables, 2

BAI_prediction, 3

calculate_BAL, 5

crownHeight_prediction, 5

data_BAI, 7

data_climate, 8

data_final_cut_weights, 9

data_ingrowth, 9

data_mortality, 10

data_NFI, 11

data_site, 12

data_tariffs, 12

data_thinning_weights, 13

data_tree_heights, 13

data_v1, 14

data_v2, 15

data_v3, 16

data_v4, 17

data_v5, 18

data_v6, 18

df_volume_parameters, 20
form_factors, 21
ingrowth_parameter_list, 23
ingrowth_table, 23
max_size_data, 24
measurement_thresholds, 24
MLFS, 25
predict_ingrowth, 31
predict_mortality, 33
simulate_harvesting, 35
volume_form_factors, 37
volume_functions, 38
volume_tariffs, 39