Package ‘sitree’

August 1, 2019

Version 0.1-7
Date 2019-07-29
Title Single Tree Simulator
Author Clara Anton Fernandez [aut, cre]
Maintainer Clara Anton Fernandez <caf@nibio.no>
Depends R (>= 3.1.0)
Imports data.table, methods, lattice, latticeExtra
Suggests knitr
VignetteBuilder knitr
Description Forecasts plots at tree level.
License GPL (>= 2)
Encoding UTF-8
LazyLoad yes
LazyData yes
NeedsCompilation no
Repository CRAN
Date/Publication 2019-08-01 12:50:02 UTC

R topics documented:

  calculate.development.class .................................................. 2
dbh.BN2009 ................................................................. 3
dead.trees.growth ............................................................. 4
fl ............................................................... 5
fn.vars.required ................................................................. 7
FuruVol ................................................................. 8
grow.dbhinc.hgtinc .......................................................... 9
height.korf ................................................................. 11
management.prob ............................................................ 12
mng.tree.removal .......................................................... 14
**calculate.development.class**

*Calculate Development Class of the Stand*

**Description**

Calculates the development class of the stand according to the Norwegian National Forest Inventory.

**Usage**

```r
calculate.development.class(SI.spp, SI.m, stand.age.years)
```

**Arguments**

- `SI.spp`: Species for which the SI is calculated (1 = spruce, 2 = pine, 3 = deciduous).
- `SI.m`: Site index (SI) in m.
- `stand.age.years`: Age of the stand in years.

**Details**

Development classes 1 (regeneration) to 5 (mature).

**Value**

It returns a vector with the development class of the stands (values 1 to 5).
**Author(s)**

Clara Anton Fernandez <caf@nibio.no>

**Examples**

```r
calculate.development.class (SI.spp = c(1, 1, 2),
SI.m = c(23, 11, 11),
stand.age.years = c(120, 80, 40))
```

---

**Description**

Implementation of the diameter at breast height function published by Bollandsås and Næsset (2009).

**Usage**

```r
dbhi.BN2009(tr, fl, common.vars, this.period, ...)
```

**Arguments**

- `tr` A `trList` class object.
- `fl` A list describing the plot data.
- `common.vars` A list with at least variables `QMD.cm`, `i.stand`, and `SBA.m2.ha`.
- `this.period` The period for which to calculate DBH increment.
- `...` Ignored

**Value**

A vector with DBH increments in mm for all trees in `tr`.

**Author(s)**

Clara Anton Fernandez <caf@nibio.no>

**References**

dead.trees.growth

Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.l <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.l,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("QMD.cm", "i.stand"),
  period.length = 5,
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31)
)

dbhi.BN2009(tr.l, fl, common.vars$res, this.period = "t0",
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31))
```

dead.trees.growth

**Growth of Dead Trees**

**Description**

A very simple function to estimate the growth of dead trees between the last measurement and their death. Dead trees are assumed to die in the middle of the period. The growth before the death of the tree is assumed to be half of the growth for the whole period, both in terms of DBH and height.

**Usage**

dead.trees.growth(dt, growth, mort, this.period)
Arguments

dt A list of trees with at least two elements: \texttt{dbh.mm} and \texttt{height.dm}.
growth A data frame with columns \texttt{dbh.inc.mm} and \texttt{hgt.inc.dm}.
mort A \texttt{TRUE}/\texttt{FALSE} vector with the predicted mortality for each living tree in this period.
this.period The period where the dead trees were last seen alive (last measured DBH).

Value

Returns a data frame of same length as \texttt{dt} with two elements

\begin{itemize}
  \item \texttt{dbh.mm} DBH in mm
  \item \texttt{height.dm} Height in dm
\end{itemize}

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

default.trees.growth(
  dt = list(dbh.mm = data.frame (t0 = c(75, 90, 25, 24)),
             height.dm = data.frame(t0 = c(190, 210, 110, 90))),
  growth = data.frame(dbh.inc.mm = c(7, 9, 2, 1),
                      hgt.inc.dm = c(11, 12, 8, 10)),
  this.period = "t0"
)

---

\texttt{fl} Plot Data

Description

Plot data

Usage

data(fl)
Format

The list contains the following elements

**plot.id** unique ID for each stand that corresponds to plot.id in the *tr* dataset

**SI.m** Site index in m

**SI.spp** Species code for which SI.m is measured

**prop.plot** Proportion of the plot covered by this stand (0-full plot, 2-20%, 3-30%, 4-40%, ..., 8-80%)

**ha2plot** multiplier to convert from "per ha" to "per plot", e.g. basal area per ha to basal area per plot

**tree2ha** multiplier to convert numbers "per tree" to "per ha", e.g. basal area per tree to basal area per ha

**soil.depth. 1234** soil depth 1(shallow)-4(deep)

**land.use** land use classification

**land.type** land type classification

**veg.type** vegetation type

**subplot.size.m2** size of the subplot corresponding to the stand in square meters

**plot.size.m2** Plot size in square meters

**kom** Municipality code

**region** region where the plot lies

**skidding.distance.100m** skidding distance in 100m

**slope.per** slope in percentage

**lat.det** latitude in degrees

**alt.m** altitude in meters

**utm.s33.ov.m** UTM E-W coordinates of the plot

**utm.s33.snv.m** UTM S-N coordinates of the plot

**stand.age.years** data frame with stand age in years, only the first column, t0, is filled up

**management** data frame with management codes for each period

Source

This dataset is derived from the Norwegian National Forest Inventory

Examples

data(fl)
str(fl)
fn.vars.required  

Finds Variables Required on Functions

Description

Tries to find the variables to be calculated through the parameters of the functions.

Usage

\[
\text{fn.vars.required}(\text{my.functions}, \ldots)
\]

Arguments

- **my.functions**: A list of functions to be used during the simulation
- **\ldots**: Use this to add any function that might not be included in my.functions

Value

It returns a vector with the name of the potential variables required.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

\[
\text{fn.vars.required (my.functions = list(}
  \text{fn.growth} = "grow.dbhinc.hgtinc",
  \text{fndbh.inc} = "dbh.B2009",
  \text{fn.hgt.inc} = "height.korf",
  \text{fn.mort} = "mort.B2007",
  \text{fn.recr} = "recr.BBG2008",
  \text{fn.management} = "management.prob",
  \text{fn.tree.removal} = "mng.tree.removal",
  \text{fn.modif} = \text{NULL},
  \text{fn.prep.common.vars} = "prep.common.vars.fun"
\text{)},
  \text{"BBG2008"})}
\]
FuruVol

Volume for the Main Species in Norway

Description

Calculates tree volume for Scots pine (FuruVol, and FuruVolV for the west coast), Spruce (GranVol, and GranVolV for the west coast), hardwoods (LauvVol), and Sitka spruce (SitkaVol) under Norwegian conditions following Braastad (1966), Brantseg(1967), Vestjordet (1967).

Usage

FuruVol(dbh, trh, bark, enhet)
FuruVolV(dbh, trh, bark, enhet)
GranVol(dbh, trh, bark, enhet)
GranVolV(dbh, trh, bark, enhet)
LauvVol(tsl, dbh, trh, bark, enhet)
SitkaVol(dbh, trh, bark, enhet)

Arguments

dbh        diameter at breast height in mm
trh        tree height in dm
bark       'ub' without bark, 'mb' with bark
enhet      units for the output. 'l' for volume in liters, and 'cm' for volume in cubic meters
tsl        tree species

Value

A vector with the individual tree volumes.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

References


Examples

FuruVol (dbh = 80, trh = 120, bark = 'ub', enhet = 'l')
FuruVolV(dbh = 80, trh = 120, bark = 'ub', enhet = 'l')
GranVol(dbh = c(80, 85), trh = c(120, 140), bark = 'ub', enhet = 'l')
GranVolV(dbh = 80, trh = 120, bark = 'ub', enhet = 'l')
LauvVol(tsl = c('30', '31'), dbh = c(80, 85), trh = c(80, 90),
    bark = 'mb', enhet = 'l')
SitkaVol(dbh = 80, trh = 120, bark = 'ub', enhet = 'l')

---

grow.dbhinc.hgtinc  \hspace{1cm} Growth and Height Increment Function Wrapper

Description

A growth and height increment function wrapper that can be used with or without a height increment function.

Usage

grow.dbhinc.hgtinc(tr, fl, common.vars, this.period, functions, ...)

Arguments

tr  A trList class object.
fl  A list describing the plot data.
common.vars  A list with at least variables spp (species classification in categories: spruce, pine, birch and other), and SBA.m2.ha.
this.period  The period for which to calculate growth.
functions  A list defining the functions to be used in sitree.
...  Further arguments to be passed to DBH increment and height increment functions.

Details

A growth function should calculate both dbh and height increment. This function calls two functions, defined in functions as fn.dbh.inc and fn.hgt.inc to estimate those.

Value

A data frame with two elements dbh.inc.mm DBH increment in mm hgt.inc.dm Height increment in dm

Author(s)

Clara Anton Fernandez <caf@nibio.no>
Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("spp", "SBA.m2.ha", "QMD.cm"),
  period.length = 5,
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31)
)

grow.dbhinc.hgtinc (tr = tr.i,
  fl = fl,
  common.vars = common.vars$res,
  this.period = "t0",
  functions = list(
    fn.growth = 'grow.dbhinc.hgtinc',
    fn.mort = 'mort.B2007',
    fn.recr = 'recr.BBG2008',
    fn.management = 'management.prob',
    fn.tree.removal = 'mng.tree.removal',
    fn.modif = NULL,  
    fn.prep.common.vars = 'prep.common.vars' ),
  fn.dbh.inc = "dbh.inc.BN2009",
  fn.hgt.inc = "height.korf",
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31)
)
```
Simple height function

Description

A simple tree height function based on tree species, and DBH. It returns the difference between the height at the current period (this.period) and the next period.

Usage

height.korf(common.vars, this.period, tr, dbh.inc.mm, ...)

Arguments

common.vars  A list with at least an element named 'spp' with the species group composition.
this.period  The period for which to calculate the height of the trees.
tr  A trList.
dbh.inc.mm  A vector with the projected dbh increment.
...  Currently not used

Value

A vector with tree heights increments in dm.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
    plot.id = tr$plot.id,
    treeid = tr$treeid,
    dbh.mm = foo.dbh,
    height.dm = foo.height,
    yrs.sim = rep(0, nrow(tr)),
    tree.sp = factor(tr$tree.sp)
  )
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
    tr = tr.i,
management.prob


definition:
"Final felling and thinning functions for Norwegian forest"

Description

Estimates the probability of a stand to be harvested or thinning following Anton-Fernandez et al. (20012).

Usage

management.prob(tr, fl, fun.final.felling = "harv.prob", fun.thinning = "thin.prob", common.vars, this.period, next.period, ...)

harv.prob(region, skidding.distance.100m, AgeTo5, vuprha.m3.ha, slope.per, SI.m, SI.spp)

thin.prob(region, skidding.distance.100m, AgeTo5, vuprha.m3.ha, slope.per, SI.m, SI.spp)

Arguments

tr
A trList class object.

fl
A list describing the plot data.

fun.final.felling
Function to use to calculate final felling.

fun.thinning
Function to use to calculate thinning.

common.vars
A list with at least variables dev.class and vuprha.m3.ha.

this.period
The period for which to calculate final felling and thinning probability.

next.period
The next period to the one for which final felling and thinning probability are to be calculated.

...
region A vector containing the region in Norway where every plot is situated.
skidding.distance.100m A vector containing skidding for each plot.
AgeTo5 A vector containing number of years to development class 5 for each plot.
vuprha.m3.ha Volume per ha in cubic meters per ha for each plot.
slope.per Slope, in percentage, for each plot.
SI.m Site index (SI) in m.
SI.spp Species for which the SI is calculated (1 = spruce, 2 = pine, 3 = deciduous).

Value

It returns a list with one element:

mng a vector with the management to apply to each plot.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

References


Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.l <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.l,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("spp", "SBA.m2.ha", "QMD.cm", "vuprha.m3.ha", "AgeTo5")
)
```
mng.tree.removal

Function to remove trees

Description

Define how trees are going to be removed in plots with some sort of management that involves removal (e.g., final felling, thinning).

Usage

mng.tree.removal(tr, fl, common.vars, this.period, next.period, ...)

---

period.length = 5,  
species.spruce = c(1, 2, 3),  
species.pine = c(10, 11, 20, 21, 29),  
species.harw = c(30, 31)

fl$management <- data.frame(matrix(NA, ncol = tr.i$nperiods,  
nrow = nrow(tr.i$data$dbh.mm)))

names(fl$management) <- paste0("t", 1:tr.i$nperiods)

management.prob(tr.i,  
fl,  
fun.final.felling = "harv.prob",  
fun.thinning = "thin.prob",  
common.vars = common.vars$res,  
this.period = "t0",  
next.period = "t1")

harv.prob(region = fl$region[1:3],  
skidding.distance.100m = fl$skidding.distance.100[1:3],  
AgeTo5 = c(50, 20, 15),  
vuprha.m3.ha = common.vars$res$vuprha.m3.ha[1:3],  
slope.per = fl$slope.per[1:3],  
SI.m = fl$SI.m[1:3],  
SI.spp = fl$SI.spp[1:3])

thin.prob(region = fl$region[1:3],  
skidding.distance.100m = fl$skidding.distance.100[1:3],  
AgeTo5 = c(50, 20, 15),  
vuprha.m3.ha = common.vars$res$vuprha.m3.ha[1:3],  
slope.per = fl$slope.per[1:3],  
SI.m = fl$SI.m[1:3],  
SI.spp = fl$SI.spp[1:3]  
)
Arguments

tr  A trList class object.
fl  A list describing the plot data.
common.vars  A list with at least variable vol.wo.tr.m3.ha
this.period  The period for which to calculate final felling and thinning probability.
next.period  The next period to the one for which final felling and thinning probability are to be calculated.

Value

Returns a TRUE/FALSE vector of length equal to the number of trees in tr.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)

tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("vol.wo.tr.m3.ha", "vuprha.m3.h"),
  period.length = 5,
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31)
)

fl$management <- data.frame(matrix(NA, ncol = tr.i$nperiods,
  nrow = length(fl$plot.id)))
names(fl$management) <- paste0("t", 1:tr.i$nperiods)

management <- management.prob(tr.i,
   fl,
   fun.final.felling = "harv.prob",
   fun.thinning = "thin.prob",
   common.vars = common.vars$res,
   this.period = "t0",
   next.period = "t1")
fl$management[, "t1"] <- management$management
remove.tree <- mng.tree.removal(tr = tr.i,
   fl = fl,
   common.vars = common.vars$res,
   this.period = "t0",
   next.period = "t1",
   per.vol.harv = 0.83 )
sum(remove.tree)

---

mort.B2007  

Mortality Function by Bollandsas (2007)

Description

Implementation of the individual tree mortality function published by Bollandsås (2007).

Usage

mort.B2007(tr, fl, common.vars, this.period, ...)

Arguments

tr  
A trList class object.

fl  
A list describing the plot data.

common.vars  
A list with at least variables spp (species classification in categories: spruce, pine, birch and other), and SBA.m2.ha.

this.period  
The period for which to calculate mortality.

...  
Ignored.

Value

A TRUE/FALSE vector for all trees in tr.

Author(s)

Clara Anton Fernandez <caf@nibio.no>
References


Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("spp", "SBA.m2.ha"),
  period.length = 5,
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31)
)

dead.trees <- mort.B2007(tr.i, fl, common.vars$res, this.period = 't0')
sum(dead.trees)
```

---

**prep.common.vars.fun**  
*Function to Calculate Common Variables*

**Description**

A function to calculate common variables needed to estimate growth, death, etc.

**Usage**

```
prep.common.vars.fun(tr, fl, i.period, this.period, common.vars, vars.required, period.length, ...)
```
Arguments

tr  A `trList` class object.
fl A list describing the plot data.
i.period The number of period (from 0 to nperiods) for which to calculate the common variables.
this.period The period for which to calculate the common variables.
common.vars A list with existing common variables.
vars.required Variables to be calculated.
period.length Length of the periods, in years.
...

Details

This is an example of a `prep.common.vars` file. A `prep.common.vars` (prepare common variables) is needed for the simulation. It is in this file that additional variables needed for the particular functions used in the simulation are calculated. For example, to calculate quadratic mean diameter, basal area, or volume.

Value

A list

res A list with the common variables.
fl Updated fl

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
```

```r
```
QMD.cm.f

tr = tr.i,
fl = fl,
i.period = 0,
this.period = "t0",
common.vars = "NULL",
vars.required = c("vol.wo.tr.m3.ha", "vuprha.m3.h"),
period.length = 5,
species.spruce = c(1, 2, 3),
species.pine = c(10, 11, 20, 21, 29),
species.harw = c(30, 31)
)
str(common.vars)

QMD.cm.f  Quadratic Mean Diameter

Description

Calculates quadratic mean diameter.

Usage

QMD.cm.f(x.mm)

Arguments

x.mm  DBH in mm.

Details

It ignores not finite values like NA, and calculates the QMD for the rest of the trees.

Value

The QMD in cm of x.mm.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

QMD.cm.f(50)
QMD.cm.f(c(51, NA))
QMD.cm.f(c(51, 42, 120, 57))
recover.last.measurement

Recover dbh and height at time of death

Description

Removes all dbh and height data that corresponds to the live of the tree, and adds the dbh and height at time of death to the dbh.mm and height.dm in the data field.

Usage

recover.last.measurement(tr.list.dead)

Arguments

tr.list.dead  a trListDead object

Details

It removes all dbh and height data from the data field, adds the dbh and height at time of death, and converts any potential NAs created in the process to 0.

Value

It returns the same object (a trListDead object) with all data on dbh and height removed. For each tree there is only one observation of dbh and height in the data field, the estimated dbh and height at time of death.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

set.seed(2017)
res <- sitree (tree.df = tr,
stand.df = fl,
    functions = list(
        fn.growth = 'grow.dbhinc.hgtinc',
        fn.mort = 'mort.B2007',
        fn.recr = 'recr.BBG2008',
        fn.management = 'management.prob',
        fn.tree.removal = 'mng.tree.removal',
        fn.modif = NULL, #'ext.modif.fun',
        fn.prep.common.vars = 'prep.common.vars.fun'
    ),
    n.periods = 5,
    period.length = 5,
    mng.options = NA,
print.comments = FALSE,
fn.dbh.inc = "dbhi.BN2009",
fn.hgt.inc = "height.korf",
species.spruce = c(1, 2, 3),
species.pine = c(10, 11, 20, 21, 29),
species.harw = c(30, 31),
fun.final.felling = "harv.prob",
fun.thinning = "thin.prob",
'BN2009',
'BBG2008', 'SBA.m2.ha', 'spp', 'pr.spru.ba', 'QMD.cm',
per.vol.harv = 0.83
)
dead <- recover.last.measurement(res$dead)
str(dead)
removed <- recover.last.measurement(res$removed)
str(removed)

recover.state Recoveres the State

Description
It adds dead and removed trees to tr. This is an internal function that should not be run, in general, by a regular user.

Usage
recover.state(tr, dead.trees, removed.trees)

Arguments
tr A trList class object.
dead.trees A trListDead class object.
removed.trees A trListDead class object.

Details
In order to make calculations easier and lighter memory wise, at each cycle (period) of the simulation all removed (harvested) and dead trees are removed from tr, which only keeps alive trees at the current period. Once the simulation is finished recover.state returns all dead and removed trees to the last tr, so sitree can return a full list, including all trees alive at any point during the simulation, of trees.

Value
A trList class object.

Author(s)
Clara Anton Fernandez <caf@nibio.no>
Recruitment function following BBG2008

Description

It creates a list of new trees based on the functions from BBG2008 (see references). BBG2008 functions estimates recruitment on a plot based on stand basal area, SI, and pine, spruce, birch, and other broadleaved species abundance. The implementation of this function is stochastic, not deterministic. So, if run several times without setting a seed, the results will vary.

Usage

```r
recr.BBG2008(tr, fl, common.vars, i.period, this.period, next.period, ...)  
BBG2008 (SBA.m2.ha, SI.m, pr.spru.ba, pr.pine.ba, pr.birch.ba, pr.other.ba)
```

Arguments

- `tr`: A trList object
- `fl`: A list describing the plot information
- `common.vars`: The number of period (from 0 to nperiods) for which to calculate recruitment.
- `i.period`: Period to use to estimate next period recruitment.
- `this.period`: Period for which to calculate recruitment.
- `next.period`: Stand basal area in m2 per ha.
- `SBA.m2.ha`: Site index (SI) in m.
- `SI.m`: Percentage of spruce in terms of basal area.
- `pr.spru.ba`: Percentage of pine in terms of basal area.
- `pr.pine.ba`: Percentage of birch in terms of basal area.
- `pr.birch.ba`: Percentage of other species in terms of basal area.
- `pr.other.ba`: Not used.

Value

Returns a data frame with two columns for each of the four species, one for the probability of positive recruitment (spruce.p, pine.p, birch.p, other.p), and one for the conditional expected recruitment (spruce.e, pine.e, birch.e, other.e).

Author(s)

Clara Anton Fernandez <caf@nibio.no>
References


Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foodbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(  
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

## create a fake dead trees and removed trees.

common.vars <- prep.common.vars.fun(  
  tr = tr.i,
  fl = fl,
  this.period = "t0",
  i.period = 0,
  common.vars = "NULL",
  mng.options = NA,
  vars.required = c("spp", "SBA.m2.ha", "pr.spru.ba"),
  period.length = 5,
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31)
)

recr <- recr.BBG2008(tr.i, fl, common.vars$res, i.period = 0, this.period = "t0",
  next.period = "t1")
str(recr)

bbg <- BBG2008 (common.vars$res$SBA.m2.ha[common.vars$res$i.tree],
  fl$SI.m,
  common.vars$res$pr.spru.ba$spru[common.vars$res$i.tree],
  common.vars$res$pr.spru.ba$pine[common.vars$res$i.tree],
  common.vars$res$pr.spru.ba$birch[common.vars$res$i.tree],
  common.vars$res$pr.spru.ba$other[common.vars$res$i.tree])
str(bbg)
```
Description

Simulates the growth of individual trees in several plots, including management and keeping record of dead and removed trees. It can also include the effect of external modifiers such as climate change.

Usage

sitree(tree.df, stand.df, functions, n.periods = 5, period.length, mng.options = NA, print.comments = FALSE,..., ext.mod.first = TRUE)

Arguments

- `tree.df`: A data frame with individual tree data. It should include four columns named 'plot.id' (an ID for the plot/stand that the tree belongs to), treeid (a unique ID for the tree), dbh (a measure of horizontal dimension), height (a measure of vertical dimension), tree.sp (tree species).
- `stand.df`: A data frame or list with necessary data on the stand (e.g. slope, site index). It should include a 'plot.id' that matches the one in tree.df and a 'plot.size.m2', the plot size in m2. 'plot.size.m2' is only used in the summary.sitree function to calculate figures by ha.
- `functions`: A list describing the basic functions. It should include a minimum of a function for growth (fn.growth), for diameter (or basal area) increment, for height increment (height.korf), mortality (fn.mort), recruitment (fn.recr), management (fn.management), tree removal (fn.tree.removal), and external modifiers (fn.modif). fn.management, fn.tree.removal, and fn.modif, can be missing.
- `n.periods`: Number of periods to simulate (numeric).
- `period.length`: The length of the periods (e.g. 5 years).
- `mng.options`: Management options and definitions to be passed further.
- `print.comments`: TRUE/FALSE Print comments. It is meant to help debugging new functions.
- `...`: Further arguments to pass to functions used during simulation.
- `ext.mod.first`: If TRUE (default) the external modifier functions will be run before fn.prep.common.vars.

Value

Returns a list with all live, dead, removed trees, and some plot information like stand age in years at each period, and management applied at each plot at each period.

Author(s)

Clara Anton Fernandez <caf@nibio.no>
Examples

```r
result.sitree <- sitree (tree.df = tr,
stand.df = fl,
functions = list(
  fn.growth = 'grow.dbhinc.hgtinc',
  fn.mort = 'mort.B2007',
  fn.recr = 'recr.BBG2008',
  fn.management = 'management.prob',
  fn.tree.removal = 'mng.tree.removal',
  fn.modif = NULL, #'ext.modif.fun',
  fn.prep.common.vars = 'prep.common.vars.fun'
),
  n.periods = 5,
  period.length = 5,
  print.comments = FALSE,
  fn.dbh.inc = "dbh.BN2009",
  fn.hgt.inc = "height.korf",
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31),
  fun.final.felling = "harv.prob",
  fun.thinning = "thin.prob",
  'BN2009',
  'BBG2008', 'SBA.m2.ha', 'spp', 'pr.spru.ba', 'QMD.cm',
  per.vol.harv = 0.83)
str(result.sitree)
set.seed(2017)
result.sitree <- sitree (tree.df = stand.west.tr,
stand.df = stand.west.st,
functions = list(
  fn.growth = 'grow.dbhinc.hgtinc',
  fn.mort = 'mort.B2007',
  fn.recr = 'recr.BBG2008',
  fn.management = NULL,
  fn.tree.removal = NULL,
  fn.modif = NULL,
  fn.prep.common.vars = 'prep.common.vars.fun'
),
  n.periods = 20,
  period.length = 5,
  mng.options = NA,
  print.comments = FALSE,
  fn.dbh.inc = "dbh.BN2009",
  fn.hgt.inc = "height.korf",
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31),
  'BN2009',
  'BBG2008', 'SBA.m2.ha', 'spp', 'pr.spru.ba', 'QMD.cm'
)
```

sitree.summary  Summary plots

Description

It produces 5 summary plots (stand basal area, number of stems, average height of tallest 10 trees, number of dead trees, and number of harvested trees)

Usage

sitree.summary(sitrees.res, plots, by.stand = TRUE, plot = FALSE, plot.all.together = FALSE)

Arguments

sitrees.res  An object result of running sitree().
plots  Plots to be printed (1: stand basal area, 2: number of stems, 3: average height of tallest 10 trees, 4: number of dead trees, 5: number of harvested trees)
by.stand  TRUE/FALSE TRUE prints plot by stand (figures by ha), if FALSE it prints plots with totals (divides values per ha by 'plot.size' and multiplies by 10.000 to obtain total values)
plot  TRUE/FALSE if plots are printed
plot.all.together  if TRUE all plots are plotted in a single page

Value

It passes the required plots in a named list. It plot is TRUE it also prints the required plots.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

See Also

sitree

Examples

require(sitree)
require(lattice)
res <- sitree (tree.df = tr,
  stand.df = fl,
  functions = list(
fn.growth = 'grow.dbhinc.hgtinc',
fn.mort = 'mort.B2007',
fn.recr = 'recr.BBG2008',
fn.management = 'management.prob',
fn.tree.removal = 'mng.tree.removal',
fn.modif = NULL,
fn.prep.common.vars = 'prep.common.vars'
),
n.periods = 5,
period.length = 5,
mng.options = NA,
print.comments = FALSE,
fn.dbh.inc = "dbhinc.BN2009",
fn.hgt.inc = "height.korf",
species.spruce = c(1, 2, 3),
species.pine = c(10, 11, 20, 21, 29),
species.harw = c(30, 31),
fun.final.felling = "harv.prob",
fun.thinning = "thin.prob",
'BN2009',
'BGG2008', 'SBA.m2.ha', 'spp', 'pr.spru.ba', 'QMD.cm',
per.vol.harv = 0.83
)

summary.plots <- sitree.summary(res, plots = 1:6, by.stand = FALSE, plot = TRUE)

sitree2dataframe

sitree2data.frame and sitree2dataframe.all

**Description**

sitree2data.frame converts trList and trListDead class objects to data.frames. sitree2data.frame converts all output of sitree to a data.frame

**Usage**

sitree2dataframe(tr.dt)
sitree2dataframe.all(sitree.res)

**Arguments**

- **tr.dt**: A trList or trListDead object.
- **sitree.res**: An object resulting from running sitree.

**Details**

As implemented, it expects to have a 'removed' element on sitree2dataframe.all. For an alternative when the simulation has no harvest see the examples below.
坐树数据框

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sitree2dataframe return a data.frame with dbh, height, and species by period. sitree2dataframe.all returns a list</td>
</tr>
<tr>
<td>live A data.frame with dbh, height, and species of the trees alive at each period.</td>
</tr>
<tr>
<td>dead A data.frame with dbh, height, and species of the trees found dead at each period.</td>
</tr>
<tr>
<td>removed A data.frame with dbh, height, and species of the trees removed at each period.</td>
</tr>
</tbody>
</table>

**Author(s)**

Clara Anton Fernandez

**Examples**

```r
n.periods <- 10

# Run a simulation with the example functions
res <- sitree (tree.df = tr,
stand.df = fl,
functions = list(
  fn.growth = "grow.dbhinc.hgtinc",
  fn.mort = "mort.B2007",
  fn.recr = "recr.BBG2008",
  fn.management = "management.prob",
  fn.tree.removal = "mng.tree.removal",
  fn.modif = NULL,
  fn.prep.common.vars = "prep.common.vars.fun"
),
n.periods = n.periods,
period.length = 5,
mng.options = NA,
print.comments = FALSE,
fn.dbh.inc = "dbhi.BN2009",
fn.hgt.inc = "height.korf",
species.spruce = c(1, 2, 3),
species.pine = c(10, 11, 20, 21, 29),
species.harw = c(30, 31),
fun.final.felling = "harv.prob",
fun.thinning = "thin.prob",
'BN2009',
'BBG2008', 'SBA.m2.ha', 'spp', 'pr.spru.ba', 'QMD.cm',
per.vol.harv = 0.83
)

dbh.mm <- sitree2dataframe(res$live)
res.dt <- sitree2dataframe.all(res)
```

## The function is currently defined as
sp.classification

Species classification

Description

Classifies the tree species into four species groups

Usage

sp.classification(tree.sp, species.spruce, species.pine, species.harw)

Arguments

tree.sp Tree species code.

species.spruce A vector with the codes used to define 'spruce' group.

species.pine A vector with the codes used to define 'pine' group.

species.harw A vector with the codes used to define the 'birch' group.

Value

Returns a vector with the classification of tree.sp into one of these four groups: spruce, pine, birch, and other. spruce group is defined by species.spruce, pine group is defined by species.pine, and birch is defined by species.harw. If a species does not fall within any of these three groups is classified as 'other'.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

sp.classification(tr$tree.sp, species.spruce = c(1,15),
                    species.pine = c(2,34),
                    species.harw = c(3,4))
stand.west.st  Stand and plot characteristics for stand.west.tr

Description

Dataset with stand and plot data for stand.west.tr

Usage

stand.west.st

Format

A data frame with 4 observations on the following 7 variables.

- plot.id  unique ID for each plot
- SI.m  site index in meters
- SI.spp species for which site index is measured
- plot.size.m2 plot size in m2
- tree2ha multiplier to convert values per tree to per hectare
- lat.deg  latitude of the plots in degrees
- subplot.size.m2 plot size in m2, this is a dummy variable needed for some of the calculations

Examples

    stand.west.st
    str(stand.west.st)

stand.west.tr  A whole stand dataset

Description

This dataset describes a whole stand of 1019 m2 and spacing 1.25x1.25m. The stand is divided in 4 plots of approximately 250 m2. All trees are Norway spruce.

Usage

    stand.west.tr
toBindLists

Format

A data frame with 651 observations on the following 5 variables.

- dbh  diameter at breast height in mm
- height  tree height in dm
- plot.id  a unique ID for each plot
- tree.id  a unique ID for each tree
- tree.sp  tree species, a factor with levels 1 2 3 10 11 20 21 29 30 31 32 40 41 42 43 44 48 49 50 51 52 53 54 55 56 57 58 59 70

Examples

```r
stand.west.tr
str(stand.west.tr)
```

Description

This function is only used internally to bind vectors, data frames, and lists.

Usage

```r
toBindLists(x, y)
```

Arguments

- `x` A vector, data frame or list.
- `y` Object of the same type as `x`.

Value

It returns a vector, a data frame or a list depending on `x`.

Author(s)

Clara Anton Fernandez <caf@nibio.no>
Examples

```r
## Example of how toBindLists works with data frames
x <- data.frame(1:10, ncol = 2)
names(x) <- c('a', 'b')
y <- data.frame(1:20, ncol = 2)
names(y) <- names(x)
j <- toBindLists(x, y)
j

## with vectors
x <- c(1:3)
y <- (8:15)
j <- toBindLists(x, y)
j

## with lists
x <- data.frame(1:10, ncol = 2)
names(x) <- c('a', 'b')
y <- data.frame(1:20, ncol = 2)
names(y) <- names(x)

x <- list(x = x, y = y)

xx <- data.frame(1:10, ncol = 3)
names(xx) <- c('a', 'b')

yy <- data.frame(1:20, ncol = 3)
names(yy) <- names(xx)

y <- list(xx = xx, yy = yy)

j <- toBindLists(x, y)
j
```

---

**tr**  
*Individual Tree Data*

Description

A data frame with data to create the trList object needed to run *sitree*.

Usage

```r
data("tr")
```

Format

The data frame contains the following columns

- **plot.id** unique ID for each stand that corresponds to plot.id in the *fl* dataset
### Description
Estimates individual tree age of trees within a plot based on basal area, site index, species, and development class.

### Usage
```r
tree.age(stand.age.years, plot.id, tree.BA.m2, dbh.mm, SI.spp, SI.m, spp, dev.class, apply.correction = TRUE)
```

### Arguments
- `stand.age.years`: The age of the stand in years.
- `plot.id`: The unique ID of the stand.
- `tree.BA.m2`: A vector with the basal areas of each tree in m2.
- `dbh.mm`: A vector with the DBH in mm of each tree.
- `SI.spp`: Species for which SI has been calculated (1, 2, 3).
- `SI.m`: Site index (SI) in m.
- `spp`: Species group classification.
- `dev.class`: Development class.
- `apply.correction`: TRUE/FALSE. If a correction to age should be applied.

### Value
A vector with the estimated ages of trees.
Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

tree.age(stand.age.years = 40,
plot.id = c(1,1), tree.BA.m2 = c(0.05, 0.5), dbh.mm
= c(50,150), SI.spp = 2,
SI.m = 11, spp = c(1,1), dev.class = 3, apply.correction = TRUE)

---

trList-class

Class "trList"

Description

Describes the Reference Class trList, which is used to store data on live trees.

Extends

All reference classes extend and inherit methods from "envRefClass".

Fields

data: Object of class list containing the unique stand ID (plot.id), the tree ID (treeid), and dbh (dbh.mm) and height (height.dm) for all the periods

nperiods: Object of class integer containing the number of period that should be simulated

Methods

extractTrees(i): Removes trees from the objectd, for example when trees die or are harvested

as.list(): converts this class to a list

show(): shows the first 20 trees

getTrees(i, j): get the information for i trees and j periods without deleting the data

addTrees(value): adds trees to the object

Note

trList objects should not be modified inside any user function. Any change should be made by the sitree function. Reference class objects are mutable, they don’t use R’s usual copy-on-modify semantics, but are modified in place. So, if you want to modify, for example for a posterior analysis of the results, any trList object I recommend to make a copy of the object first, using $copy(), e.g. my.trListCopy().
Author(s)

Clara Anton Fernandez <caf@nibio.no>

See Also

trListDead

Examples

showClass("trList")

---

trListDead-class  Class "trListDead"

Description

A Reference Class for dead trees.

Extends

Class "trList", directly.

All reference classes extend and inherit methods from "envRefClass".

Fields

data: Object of class list containing the unique stand ID, the tree ID, and dbh and height for all the periods. Only data from the periods were the tree was alive is stored in this field

nperiods: Object of class integer containing the number of period that should be simulated

last.measurement: Object of class data.frame containing tree ID, dbh, height, and period when the tree was "found" dead

Methods

last.time.alive(): It estimates when the trees where last "seen" alive

addTrees(value): It add trees to the trlistDead object

remove.next.period(next.period): It removed the data from the period when the tree was seen "dead". The next period is calculated so the dbh and height at time of death can be estimated

The following methods are inherited (from the corresponding class): extractTrees ("trList"), as.list ("trList"), show ("trList"), getTrees ("trList"), addTrees ("trList")

Note

This class is meant to keep the data of the dead trees in a similar way as the live trees, but with some extra information such as diameter and height at death.
Author(s)

Clara Anton Fernandez <caf@nibio.no>

See Also

trList, ~~~

Examples

showClass("trListDead")
Index

+ Topic **classes**
  trList-class, 34
  trListDead-class, 35
+ Topic **datasets**
  fl, 5
  stand.west.st, 30
  stand.west.tr, 30
  tr, 32

BBG2008 (recr.BBG2008), 22

calculate.development.class, 2

dbhi.BN2009, 3
dead.trees.growth, 4

evRefClass, 34, 35

f1, 5
fn.vars.required, 7
FuruVol, 8
FuruVolV(FuruVol), 8
GranVol(FuruVol), 8
GranVolV(FuruVol), 8
growdbhinc.hgtinc, 9

harv.prob (management.prob), 12
height.korf, 11
LauvVol(FuruVol), 8
management.prob, 12
mng.tree.removal, 14
mort.B2007, 16

prep.common.vars.fun, 17
QMD.cm.f, 19

recover.last.measurement, 20
recover.state, 21

recr.BBG2008, 22
SitkaVol(FuruVol), 8
sitree, 24, 26
sitree.summary, 26
sitree2dataframe, 27
sp.classification, 29
stand.west.st, 30
stand.west.tr, 30

thin.prob (management.prob), 12
toBindLists, 31
tr, 32
tree.age, 33
trList, 35, 36
trList (trList-class), 34
trList-class, 34
trListDead, 35
trListDead (trListDead-class), 35
trListDead-class, 35