Package ‘forestat’

May 25, 2023

Type Package
Title Evaluating the Quality of Natural Forest
Version 1.0.1
Description Including tree height classification, tree height model establishment, sectional area growth and stock growth modeling, and calculation of actual and potential forest productivity.
Maintainer Yuanyuan Han <jackhanyuan@foxmail.com>
Repository CRAN
BugReports https://github.com/caf-ifrit/forestat/issues
License GPL (>= 3)
Encoding UTF-8
Language en-US
LazyData true
Depends R (>= 2.10)
RoxygenNote 7.2.3
Imports dplyr, ggplot2, graphics, nlme, stats
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Liyong Fu [aut],
    Shirong Liu [aut],
    Shouzheng Tang [aut],
    Guangshuang Duan [aut],
    Zhong Wang [aut],
    Linyan Feng [aut],
    Dongbo Xie [aut],
    Yuanyuan Han [aut, cre]
Date/Publication 2023-05-25 09:10:02 UTC
R topics documented:

class.plot .......................................................... 2
forestData .......................................................... 3
plot.forestData ...................................................... 4
potential.productivity ............................................. 5
reality.productivity ................................................ 6
summary.forestData ............................................... 7

Index

class.plot Calculate tree height grading parameters

Description
class.plot adds new variables: the original tree height grading and the adjusted tree height grading.
And the existing variables are retained.

Usage
class.plot(
data,  
model = "Logistic",  
interval = 5,  
number = 5,  
a = 30,  
b = 10,  
c = 0.5,  
maxiter = 1000
)

Arguments
data A data.frame data in which at least four columns are required as input: ID, code, AGE, H.
model Type of model used for building the H model (tree height model), options are ‘Logistic’, ‘Richards’, ‘Korf’, ‘Gompertz’, ‘Weibull’, or ‘Schumacher’.
interval The intervals for tree height classification.
number The maximum value of the tree height classification interval.
a, b, c The initial parameters of the fitted model.
maxiter The maximum number of iterations to fit the model.

Details
Input takes a data.frame with three variables ID, AGE, H and returns the tree height grading value of every sample(rows in the data.frame).
**Value**

A data of forestData class with output values, models and model parameters.

**Examples**

```r
# Load sample data
data("forestData")

# Build a model based on the forestData and return a forestData class object
forestData <- class.plot(forestData,model="Richards",
                          interval=5,number=5,
                          a=19,b=0.1,c=0.8)
```

---

**Description**

Subzone 6 Oak broadleaf mixed data from fieldwork

**Usage**

```r
forestData
```

**Format**

A data frame with 1,326 rows and 79 columns:

- **ID** Unique identifier for each tree
- **AGE**  Stand age of the tree
- **H**  Height of the tree
- **BA** Basal area of the tree
- **Bio** Biomass of the tree
- **S** Forest density index
- **code**  Codes for forest types ...
plot.forestData  

ForestData Plot  

Description  
Plot graphs about the forestData.

Usage  
```r  
## S3 method for class 'forestData'
plot(
  x,
  model.type = "H",
  plot.type = "Curve",
  xlab = NA,
  ylab = NA,
  legend.lab = "Site class",
  title = "Oak broadleaf mixed",
  ...
)
```

Arguments  
- `x` A data of forestData class.
- `model.type` Type of model used for fitting. options are ‘H’ (tree height model), ‘BA’ (basal area growth model), or ‘Bio’ (stocking growth model).
- `plot.type` Type of plot, options are ‘Curve’ (curve plot), ‘Scatter_Curve’ (scatter plot with curve), ‘residual’ (residual plot), or ‘Scatter’ (scatter plot).
- `xlab` The title for the x axis.
- `ylab` The title for the y axis.
- `legend.lab` The title for the legends.
- `title` The text for the Plot title.
- `...` Additional arguments affecting the figure plotted.

Value  
A trellis plot object

Examples  
```r  
# Load sample data
data("forestData")

# Build a model based on the forestData and return a forestData class object
forestData <- class.plot(forestData,model="Richards",
```
potential.productivity

\[
\text{interval}=5, \text{number}=5, \\
a=19, b=0.1, c=0.8)
\]

# Plot the curve of the tree height model
plot(forestData, model.type="H", \\
plot.type="Curve", \\
xlab="Stand age (year)", ylab="Height (m)", legend.lab="Site class", \\
title="Curve of the Oak and Broadleaf Tree Height Model")

---

potential.productivity

*Calculate the potential productivity.*

**Description**

potential.productivity calculate the potential productivity of each tree based on model parameters obtained from the parameterOutput function.

**Usage**

```r
potential.productivity(
  forestData, 
  code = 1, 
  age.min = 5, 
  age.max = 150, 
  left = 0.05, 
  right = 100, 
  e = 1e-05, 
  maxiter = 50
)
```

**Arguments**

- **forestData** A forestData class data
- **code** Codes for forest types.
- **age.min** The minimum age of production potential.
- **age.max** The maximum age of production potential
- **left** Solving for the left boundary of the potential productivity.
- **right** Solving for the right boundary of the potential productivity.
- **e** Accuracy parameters for solving the forest density index according to Newton’s iterative method.
- **maxiter** Maximum number of iterations parameter for solving the forest density index according to Newton’s iteration method.
potential.productivity takes data_BA, data_V parameters as required inputs.

A forestData class in which a data.frame with potential productivity parameters is added.

Examples

```r
# Load sample data
data("forestData")

# Build a model based on the forestData and return a forestData class objectorestData <- class.plot(forestData, model="Richards,"
interval=5, number=5,
a=19, b=0.1, c=0.8)

# Calculate the potential productivity of the forestData object
forestData <- potential.productivity(forestData, code=1,
age.min=5, age.max=150,
left=0.05, right=100,
e=1e-05, maxiter=50)
```

reality.productivity  

Calculate the reality productivity.

Description

reality.productivity calculate the reality productivity of each tree based on model parameters (obtained from the parameterOutput function).

Usage

```r
reality.productivity(forestData, left = 0.05, right = 100)
```

Arguments

- forestData: A forestData class data
- left: Solving for the left boundary of the reality productivity.
- right: Solving for the right boundary of the reality productivity.

Details

reality.productivity takes data, data_BA, data_V parameters as required inputs.
Value

A forestData class in which a data.frame with reality productivity parameters is added.

Examples

```r
# Load sample data
data("forestData")

# Build a model based on the forestData and return a forestData class object
forestData <- class.plot(forestData,model="Richards",
interval=5,number=5,
a=19,b=0.1,c=0.8)

# Calculate the reality productivity of the forestData object
forestData <- reality.productivity(forestData,left=0.05,right=100)
```

---

summary.forestData  
Summary of forestData

Description

Generates summary statistics for forestData objects.

Usage

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

Arguments

- `object`  
  A forestData object (after class.plot).
- `...`  
  Additional arguments affecting the summary produced.

Details

The summary includes the summary of raw data, the model, the model parameters, potential productivity and real productivity in forestData(if available)

Value

A summary object of class "summary.forestData"
Examples

# Load the forestat.csv sample data
forestData <- read.csv(system.file("extdata", "forestData.csv", package = "forestat"))

# Build a model based on the forestData and return a forestData class object
forestData <- class.plot(forestData,model="Richards",
                          interval=5,number=5,
                          a=19,b=0.1,c=0.8)

# Get the summary data of the forestData object
summary(forestData)
Index

* datasets
  forestData, 3

class.plot, 2

forestData, 3

plot.forestData, 4

potential.productivity, 5

reality.productivity, 6

summary.forestData, 7