# Package ‘treecm’

February 20, 2015

**Type** Package

**LazyLoad** yes

**License** GPL-2

**Title** Centre of mass assessment and consolidation of trees.

**Author** Marco Bascietto <marco.bascietto@cnr.it>

**Maintainer** Marco Bascietto <marco.bascietto@cnr.it>

**Description** The centre of mass is a crucial data for arborists in order to consolidate a tree using steel or dynamic cables. Given field-recorded data on branchiness of a tree, the package: (i) computes and plots the centre of mass of the tree itself, (ii) computes branches slenderness coefficient in order to aid the arborist identify potentially dangerous branches, and (iii) computes the force acting on a ground plinth and its best position relating to the tree centre of mass, should the tree need to be stabilized by a steel cable

**Version** 1.2.1

**URL** https://github.com/mbask/treecm

**Date** 2014-05-22

**Imports** plyr

**Suggests** ggplot2, reshape2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-05-22 12:18:18

## R topics documented:

---

`treecm-package` *Assessment of the position of the centre of mass of trees*

---

## Description

The centre of mass is a crucial data for arborists in order to consolidate a tree using steel or dynamic cables. Given field-recorded data on branchiness of a tree, the package:

- computes and plots the centre of mass of the tree itself
- simulates the shift in CM position as branches are pruned
- computes branches slenderness coefficient in order to aid the arborist identify potentially dangerous branches

- computes the force acting on a ground plinth and its best position relating to the tree centre of mass, should the tree need to be stabilized by a steel cable

The tree stem is ideally sectioned in logs. The weight of tree components is assessed based on

- the sum of volume of stem logs
- the sum of branches biomass

Field measures to be taken on logs and branches are described in `importFieldData` and are to be recorded on the tree itself, possibly using tree-climbing tecniques. In order to help the arborist in the pruning selection process a simple plot of branch coefficient of slenderness is implemented.

## Note

**Branch biomass** is computed by allometric equations relating its fresh weight (wood + leaves) to its diameter at point of insertion on the stem. **Log biomass** is computed by converting its volume to weight using wood fresh density. Volume is computed using Smalian's formula (see `logBiomass` description). A sample `.CSV` file is provided to guide through data filling in the field

## Author(s)

Marco Bascietto `<marco.bascietto@cnr.it>`

## References

Source code is hosted at GitHub (`http://mbask.github.com/treecm/`)

## See Also

`logBiomass` `importFieldData`

## Examples

```
data(stonePine1TreeData)
vectors  <- treeVectors(stonePine1TreeData)
CM       <- centreOfMass(vectors)
plot(vectors,
   main = "Centre Of Mass",
   col = "grey30",
   txtcol = "grey30")
plot(CM)
summary(CM)
```

`allometryABDC` *Returns the fresh weight of a stone pine branch*

## Description

Returns the fresh biomass of a stone pine branch in kg given the diameter, using an allometric equation

## Usage

```
allometryABDC(x, diameter)
```

## Arguments

| | |
|---|---|
| `x` | a data frame holding diameters of branches |
| `diameter` | the name of the column holding diameter of the x data frame, diameters should be in cm |

## Value

the fresh biomass of the branch of a stone pine (in kg)

## Note

The allometric equation has been validated for 5-16 cm diameter branches.

The allometric equation takes the form of a power equation. This equation yields more correct results than `allometryAsca2011` since it has been built on a wider range of branch diameters and it superseeds it.

## References

Data collected by A. Ascarelli and integrated by small diameter branches by M. Bascietto and B. De Cinti, non linear regression by M. Bascietto

## See Also

`powerEquation`

Other Biomass: `allometryAsca2011`; `allometryCutini2009`; `allometryPorte2002`; `logBiomass`; `powerEquation`; `pureQuadraticEquation`; `treeBiomass`; `treeTotalBiomass` `allometryAsca2011` *Returns the fresh weight of a stone pine branch*

## Description

Returns the fresh biomass of a stone pine branch in kg given the diameter, using an allometric equation

## Usage

```
allometryAsca2011(x, diameter)
```

## Arguments

| | |
|---|---|
| `x` | a data frame holding diameters of branches |
| `diameter` | the name of the column holding diameter of the x data frame, diameters should be in cm |

## Value

the fresh biomass of the branch of a stone pine (in kg)

## Note

The allometric equation has been validated for 8-16 cm diameter branches.

The allometric equation takes the form of a power equation

## References

Data collected by A. Ascarelli, non linear regression by M. Bascietto

## See Also

`powerEquation`

Other Biomass: `allometryABDC`; `allometryCutini2009`; `allometryPorte2002`; `logBiomass`; `powerEquation`; `pureQuadraticEquation`; `treeBiomass`; `treeTotalBiomass` allometryCutini2009 *Returns the biomass of a stone pine tree*

## Description

Returns total biomass of a stone pine tree (wood and leaves, dry state) in kg given the diameter at breast height, using an allometric equation

## Usage

```
allometryCutini2009(x, diameter)
```

## Arguments

| | |
|---|---|
| `x` | a data frame holding diameters of branches |
| `diameter` | the name of the column holding diameter of the x data frame, diameters should be in cm |

## Value

the total biomass of the branch of a stone pine (in kg, dry state)

## Note

Use this function at you own risk, it has been validated for trees (ie: >24 cm diameters).

The allometric equation takes the form of a pure quadratic equation

## References

Cutini, A. and Hajny, M. and Gugliotta, O. and Manetti, M. and Amorini, E. 2009, Effetti della struttura del popolamento sui modelli di stima del volume e della biomassa epigea (Pineta di Castelfusano - Roma) *Forest@*, **6**, 75–84 Tipo B

## See Also

`pureQuadraticEquation`

Other Biomass: `allometryABDC`; `allometryAsca2011`; `allometryPorte2002`; `logBiomass`; `powerEquation`; `pureQuadraticEquation`; `treeBiomass`; `treeTotalBiomass`

## allometryPorte2002 *Returns the biomass of a maritime pine branch*

### Description

Returns the woody biomass of a maritime pine branch (dry state, no leaves!) in kg given the diameter, using an allometric equation

### Usage

```
allometryPorte2002(x, diameter)
```

### Arguments

| | |
|---|---|
| `x` | a data frame holding diameters of branches |
| `diameter` | the name of the column holding diameter of the x data frame, diameters should be in cm |

### Value

the woody biomass (dry state, no leaves!) of the branch of a maritime pine (in kg)

### Note

The allometric equation has been validated for <10 cm diameter branches, extrapolation on larger branches my yield unreasonable results.

The allometric equation takes the form of a power equation

### References

Port\'e, A. and Trichet, P. and Bert, D. and Loustau, D. 2002, Allometric relationships for branch and tree woody biomass of Maritime pine (*Pinus pinaster* Ait.) *Forest Ecology and Management*, **158**, 71–83

### See Also

`powerEquation`

Other Biomass: `allometryABDC`; `allometryAsca2011`; `allometryCutini2009`; `logBiomass`; `powerEquation`; `pureQuadraticEquation`; `treeBiomass`; `treeTotalBiomass` anchorRange *Min/max values for the anchor position along the stem*

## Description

The anchor level must not be lower than the tree CM (for obvious static reasons) and higher than "main stem" height

## Usage

```
anchorRange(logs, CM)
```

## Arguments

| | |
|---|---|
| logs | a data frame holding the selected logs (see logPathSelection) |
| CM | an object of CM class |

## Value

a named vector of 2 elements:

| | |
|---|---|
| z | the height of the CM |
| hMax | the height of the "main stem" tip |

## See Also

Other Stabilization: centreOfMassAngle; centreOfMassModulus; getPlinthForce; logPathSelection; toCartesianXYZ

## Examples

```
library(treecm)
data(stonePine1TreeData)
vectors <- treeVectors(stonePine1TreeData)
CM <- centreOfMass(vectors)
logs <- logPathSelection(stonePine1TreeData)
anchorRange(logs, CM)
```

`branchSR` *Compute the slenderness ratio*

## Description

slenderness ratio is an important index of stability of trees and branches

## Usage

```
branchSR(x, diameter, length, tilt)
```

## Arguments

| | |
|---|---|
| `x` | the data frame holding the measures needed to perform the computation |
| `diameter` | The name of the data frame column holding diameter of the branch |
| `length` | The name of the data frame column holding length of the branch |
| `tilt` | The name of the data frame column holding tilt of the branch |

## Value

slenderness ratio

## Note

The coefficient takes into account branch angle: $SL_c = \frac{L}{D} \cdot (1 + cos\alpha)$, where $\alpha$ is the branch angle (0 degrees = horizontal, 90 degrees vertical), $L$ is branch length in m, $D$ is branch diameter in cm Vertical branches have $SL = SL_c$

## References

Mattheck, C. and Breloer, H. *The Body Language of Trees: A Handbook for Failure Analysis (Research for Amenity Trees)* 1995, HMSO (London)

`buildMomentObject` *Constructor for the generic class moment*

## Description

Create an instance of a moment class holding mass, arm length and angle between the arm and the weight vector

## Usage

```
buildMomentObject(length, mass, angle)
```

### Arguments

| | |
|---|---|
| `length` | the length of the arm of the moment |
| `mass` | the mass of the moment (in kg) |
| `angle` | the angle between the moment arm and the vector pointing towards the ground (the weight vector) |

### Value

an instance of `moment` and list classes

### See Also

Other Stabilization momentClass: `buildTreeMomentObject`; `calcMoment`; `getMoment`

---

## `buildTreeMomentObject` *Constructor for the generic class treeMoment*

---

### Description

The class inherits from `moment` without adding any more properties

### Usage

```
buildTreeMomentObject(length, mass, angle)
```

### Arguments

| | |
|---|---|
| `length` | the distance from tree base to CM, as computed by `centreOfMassModulus` |
| `mass` | the tree mass (in kg), not its weight, as computed by `treeTotalBiomass` |
| `angle` | the angle between the moment arm (from tree base to CM) and the vector pointing towards the ground (the tree weight vector), as computed by `centreOfMassAngle` |

### Value

an instance of `treeMoment`, `moment` and list classes

### See Also

`calcMoment`

Other Stabilization momentClass: `buildMomentObject`; `calcMoment`; `getMoment`

## calcMoment — *Computes moment and returns the moment object*

### Description

Moment is computed as $M = l \cdot F$, where l is moment arm, F is the component of the force (*mass times g*) normal to moment arm

### Usage

```
calcMoment(object, g = 9.81)
```

### Arguments

| | |
|---|---|
| `object` | an instance of moment class |
| `g` | the standard gravity |

### Value

the updated moment object

### See Also

`getPlinthForce`

Other Stabilization momentClass: `buildMomentObject`; `buildTreeMomentObject`; `getMoment`

## centreOfMass — *Computes the centre of mass of the tree*

### Description

The $x$ coordinate of the centre of mass is defined as $\frac{\sum(m_i x_i)}{\sum(m_i)}$ where $m_i$ is the biomass of the $i^{th}$ branch and $x_i$ is the $x$ coordinate of the $i^{th}$ branch. $y$ and $z$ coordinates are similarly computed. The centre of mass computation excludes branches to be pruned (ie: those whose `toBePruned` value is set to `TRUE`).

### Usage

```
centreOfMass(object)
```

### Arguments

| | |
|---|---|
| `object` | A data frame of class `vectors` |

### Value

A vector holding $x$, $y$, $z$ coordinates of the centre of mass

## centreOfMassAngle *Returns the angle between CM modulus and the tree weight vector*

### Description

Returns the angle between CM modulus and the tree weight vector

### Usage

```
centreOfMassAngle(object)
```

### Arguments

| | |
|---|---|
| `object` | an object of `CM` class |

### Value

a real number in radians

### Note

This function is mainly needed to compute the moment of the tree. The angle is need to compute the projection of the tree weight normal to the CM modulus

### See Also

Other Stabilization: `anchorRange`; `centreOfMassModulus`; `getPlinthForce`; `logPathSelection`; `toCartesianXYZ`

### Examples

```
library(treecm)
data(stonePine1TreeData)
vectors <- treeVectors(stonePine1TreeData)
CM <- centreOfMass(vectors)
print(centreOfMassAngle(CM))
treeMoment <- buildTreeMomentObject(
 centreOfMassModulus(CM)
 , treeTotalBiomass(stonePine1TreeData)
 , centreOfMassAngle(CM)
 )
```

## centreOfMassModulus *Computes the modulus of the CM vector*

### Description

The Centre of Mass vector starts from the tree base and points towards the CM. Its modulus is the distance between the CM (x, y, z) and tree base (0, 0, 0).

### Usage

```
centreOfMassModulus(object)
```

### Arguments

| | |
|---|---|
| `object` | an object of `CM` class |

### Value

a real number

### Note

This function is mainly needed to compute the moment of the tree. The CM modulus is the tree moment arm.

### See Also

Other Stabilization: `anchorRange`; `centreOfMassAngle`; `getPlinthForce`; `logPathSelection`; `toCartesianXYZ`

### Examples

```
library(treecm)
data(stonePine1TreeData)
vectors <- treeVectors(stonePine1TreeData)
CM <- centreOfMass(object=vectors)
print(centreOfMassModulus(CM))
treeMoment <- buildTreeMomentObject(
 centreOfMassModulus(CM)
 , treeTotalBiomass(stonePine1TreeData)
 , centreOfMassAngle(CM)
 )
```

`Dst` *Green wood density data for a few tree species*

### Description

Wood density is used to convert wood volume measures in the field to their fresh weights. It is measured in $\frac{kg}{m^3}$.

### Format

A data frame with 170 observations on the following 3 variables. `data.frame: 170 obs. of 3 variables:`

```
$ species: chr  "Abies alba" "Abies alba" "Abies balsama" "Abies grandis" ...
$ group  : Factor w/ 2 levels "conifer","dicot": 1 1 1 1 1 1 1 1 1 1 ...
$ density: int  545 577 529 449 465 673 689 497 673 577 ...
```

### Details

Density is measured at humidity level 50 wood humidity. The dataset is provided as a reference only, please be cautioned about using these values on your samples.

### Source

Niklas, K. J. and Spatz, H.-C. (2010) Worldwide correlations of mechanical properties and green wood density. American Journal of Botany, 97, 1587-1594

### Examples

```
data(Dst)
```

`getCoordinatesAndMoment` *Returns the coordinates of centre of mass of branches and logs*

### Description

Computes the cartesian coordinates of centre of mass of branches and logs along with their $x$, $y$, $z$ moments

The $x$ and $y$ coordinates are computed from the polar coordinates (angle and distance, defined as the length of its projection on ground), measured in the field. The $z$ coordinate is computed by adding the height of branch insertion on the stem (measured in the field) to the height of the branch (calculated through its mean tilt, in case it was measured in the filed). The $x$, $y$, $z$ coordinates are corrected to take into account where the actual centre of mass lies on the branches themselves by multiplying them by branchesCM, a real number from 0.01 (CM at branch base) to 1.00 (CM at branch tip). As a rule of thumb, average live branches, with an average amount of foliage, have CM approx. $2/3$ of their length, ie. branchesCM = 0.66. $x$, $y$, $z$ moments are computed by multiplying the corresponding cartesian coordinates by branch or log mass, e.g. $m_x = F \cdot l_x$, where $F$ is branch or log mass, $l_x$ is the $x$ component of the lever arm (e.g. the $x$ component of the branch or log projection on the ground).

## Usage

```
getCoordinatesAndMoment(azimuth, dBase, dTip, length, tipD, height, tilt,
  toBePruned, biomass, branchesCM)
```

## Arguments

| | |
|---|---|
| azimuth | Branch compass heading |
| dBase | unused argument |
| dTip | unused argument |
| length | Branch length |
| tipD | unused argument |
| height | Height of branch insertion on the stem or the height of log lower section |
| tilt | Inclination of the branch or log in degrees |
| toBePruned | unused argument |
| biomass | Mass of the branch or log |
| branchesCM | a real number varying from 0.01 to 1 proportional to the centre of mass position along the branch (0.01 branch base, 1 branch tip) |

## Value

a vector holding 5 reals:

- the $x$ coordinate of branch CM
- the $y$ coordinate of branch CM
- the $x$ moment of the branch
- the $y$ moment of the branch
- the $z$ moment of the branch

## Note

BranchCM is assumed to have same value in branches and logs. This is not the case in the real world. As a measure of safety one should use the highest value possible, eg branchesCM = 1.

$z$ coordinate of CM is not returned because it would be useless in a 2D plot. It is computed using $mz$, which is, as a matter of facts, returned

## getMoment — *Returns the moment*

### Description

For a description of how moment is computes see calcMoment. The function computes moment, if not yet done in a previous user call to calcMoment, and only returns it without updating the moment object

### Usage

```
getMoment(object)
```

### Arguments

| | |
|---|---|
| object | an instance of moment class |

### Value

the moment figure

### See Also

getPlinthForce

Other Stabilization momentClass: buildMomentObject; buildTreeMomentObject; calcMoment

## getPlinthForce — *Computes the force on the plinth on the ground*

### Description

To stabilize the tree a steel cable is connected from an anchor point on the tree to a plinth on the ground. The function computes the force on the plinth (needed to choose the appropriate steel cable and to build the plinth itself) and the maximum security azimuth (the angle relative to the North from the tree base). Force is computed by comparing the moment of the tree and the moment of the anchor, whose arm is the vector from tree base to the anchor point. The anchor point is defined as the distance from tree base, along the stem. Note that this distance equals anchor height only when the stem is perfectly vertical and straight.

### Usage

```
getPlinthForce(l.stem, d, logs, treeMoment, CM)
```

## Arguments

| | |
|---|---|
| `l.stem` | the distance from tree base to anchor point, along the stem |
| `d` | the length of the cable (in metres) |
| `logs` | a data frame holding the selected logs (see `logPathSelection`) |
| `treeMoment` | the moment of the tree as computed by `calcMoment` |
| `CM` | an object of `CM` class |

## Value

a named list of 6 elements:

| | |
|---|---|
| `force` | the force in Newton on the plinth |
| `distanceOnGround` | the distance from tree base to the plinth |
| `anchorAlongStem` | the distance from tree base to the anchor (*ie* `l.stem`) |
| `cableLength` | the length of the cable (*ie* `d`) |
| `anchorHeight` | true height of the anchor over ground |
| `azimuth` | the azimuth of the plint relative to the tree base |

## Note

The function is vectorized both for anchor distance from tree base (`l.stem` parameter) and for cable length (`d`). It is not possible to pass invalid `l.stem` values, see `anchorRange`.

## See Also

Other Stabilization: `anchorRange`; `centreOfMassAngle`; `centreOfMassModulus`; `logPathSelection`; `toCartesianXYZ`

## Examples

```
library(treecm)
data(stonePine1TreeData)
vectors <- treeVectors(stonePine1TreeData)
CM <- centreOfMass(vectors)
treeMoment <- buildTreeMomentObject(
  centreOfMassModulus(CM)
  , treeTotalBiomass(stonePine1TreeData)
  , centreOfMassAngle(CM)
  )
treeMoment <- calcMoment(treeMoment)
logs <- logPathSelection(stonePine1TreeData)
plinth <- data.frame(getPlinthForce(10, 20, logs, getMoment(treeMoment), CM))
```

`importFieldData` *Imports field data from csv file*

## Description

Imports `csv` file holding field recorded data returning a list holding field and other key data provided as arguments.

Field measures to be taken on logs and branches include:

- **Azimuth** (`azimuth`): mean angle of orientation of the branch or log measured from the base of the tree (usually with magnetic north as reference, measured clockwise), *mandatory*
- **Diameter at base** (`dBase`): diameter at insertion point on the stem, for branches, diameter of the lower section for logs, *mandatory*
- **Diameter at top** (`dTip`): always 0 for branches, diameter of the higher section for logs, defaults to 0, *mandatory* only for logs
- **Length** (`length`): length of logs or branches, *mandatory* for logs, *mandatory* for branches only if their slenderness coefficient is to be computed
- **Distance** (`tipD`): length of branch or log projection on the ground, starting from tree base to tip of branch or log, *mandatory*
- **Height** (`height`): height of branch insertion on the stem or height of lower section of the log, to be used to compute `z` coordinate of CM, defaults to NA, *mandatory* only for `z` determination of CM
- **Tilt** (`tilt`): mean branch tilt or log tilt from the horizontal plane (eg a vertical branch is 90 degrees, an horizontal branch is 0 degrees), to be used to compute `z` coordinate of CM, defaults to 0, *mandatory* only for `z` determination of CM. Note however that the tree tip should be considered as a branch, not a log, in order to account for foliage biomass. In this case the tilt value should be recorded otherwise it would default to 0, *ie* an horizontal branch
- **To be pruned** (`toBePruned`): a boolean value, defaults to FALSE, *optional*
- **Path to tip** (`pathToTip`): a boolean value TRUE on logs and branches that make part of the "main stem" of the tree, defaults to FALSE, *optional*

## Usage

```
importFieldData(fileName, dst, branchesAllometryFUN, bCM = 1)
```

## Arguments

| | |
|---|---|
| `fileName` | Name of csv file holding field data |
| `dst` | Fresh density of wood of the tree |
| `branchesAllometryFUN` | the function that should compute branch biomass from its diameter |
| `bCM` | Estimated position of the centre of mass of branches, a real number from 0.01 (CM at branch base) to 1.00 (CM at branch tip). As a rule of thumb, average live branches, with an average amount of foliage, have CM approx. from $1/3$ to $2/3$ of their length. bCM = 1.0 (default value) |

### Value

a list of 4 elements: field data, wood fresh density, allometryFUN function and branches CM

### See Also

getCoordinatesAndMoment

---

`logBiomass` *Estimates the wood biomass of logs and truncated branches*

---

### Description

Estimates the wood biomass of logs and truncated branches by computing their volume (using Smalian's formula) and converting it to fresh weight using wood fresh density.

Smalian's formula: $V = \frac{Sb+Sd}{2}l$ where $V$ is the log volume, $Sb$ is the aerea of the basal (lower) section, $Sd$ is the area of the higher section and $l$ is the length of the log.

### Usage

```
logBiomass(x, lowerD, higherD, logLength, density)
```

### Arguments

| | |
|---|---|
| `x` | the data frame holding the measures needed to perform the estimation |
| `lowerD` | The name of the data frame column holding diameter of the lower section in cm |
| `higherD` | The name of the data frame column holding the diameter of the higher section (usually smaller!) in cm |
| `logLength` | The name of the data frame column holding the length of the log or branch in m |
| `density` | The name of the data frame column holding the fresh density of the wood, defined as $D = \frac{V_f}{W_f}$ where $V_f$ is wood volume measured in the field (i.e. satured with water) in $m^3$ and $W_f$ is wood fresh weight in kg. Fresh density is measured in $\frac{kg}{m^3}$ |

### Note

Diameters used to compute section areas should be measured under the bark layer! When this is not the case (scarcely ever!) and diameters include bark thickness the log biomass is somewhat over-estimated!

### References

la Marca, O. *Elementi di dendrometria* 2004, Patron Editore (Bologna), p. 119

### See Also

Other Biomass: allometryABDC; allometryAsca2011; allometryCutini2009; allometryPorte2002; powerEquation; pureQuadraticEquation; treeBiomass; treeTotalBiomass

## logPathSelection — *Returns tree logs and branches being part of the "main stem"*

### Description

The "main stem" is not very clearly defined. Most softwood species (*ie Picea*, *Abies*, *Pseudotsuga* etc.) have only one stem given by their dominant apex. Hardwood species (*Quercus*, *Tilia* etc) and some softwood species (*eg Pinus pinea*) do not exhibit a dominant apex and branches often enlarge and grow taller than the apex. In these latter cases one has to select an appropriate path from tree base to tree tip, according to what may be considered the "main stem" of the tree. Both in former and latter cases the path selection has to be laid down in the `fieldData` data frame.

### Usage

```
logPathSelection(treeData)
```

### Arguments

| | |
|---|---|
| `treeData` | A named list that includes a `fieldData` data frame element, holding `pathToTip`-, `azimuth`-, `length`-, `tilt`-named columns |

### Value

a data frame subsetted from the `fieldData` data frame having `TRUE` selected branches and logs, with three columns: `azimuth`, `length`, `tilt`. The first row if filled with zeros.

### Note

Selected branches and logs have a `TRUE` value in the `pathToTip` column. This is a necessary step towards anchor force detemrination, as the returned data frame has to be submitted to `getPlinthForce`

### See Also

`importFieldData`

Other Stabilization: `anchorRange`; `centreOfMassAngle`; `centreOfMassModulus`; `getPlinthForce`; `toCartesianXYZ`

### Examples

```
library(treecm)
data(stonePine1TreeData)
logs <- logPathSelection(stonePine1TreeData)
```

## plot.CM *Plots tree CM*

### Description

Plots tree centre of mass as a layer on top of the `plot.vector`.

CMs vector radii are proportional to CM magnitude. Tree CM is connected to tree base by an arrow showing the direction the tree would take in case of it falling down. $z$ coordinate of tree CM is printed alongside its vector (if branch height has been recorded in the field).

### Usage

```
## S3 method for class 'CM'
plot(x, y = NULL, ...)
```

### Arguments

| | |
|---|---|
| `x` | CM object |
| `y` | unused |
| `...` | Arguments to be passed to plot.default |

### Value

`NULL`

## plot.SR *Plots slenderness ratio of branches*

### Description

Plots the branches as arrows whose length is proportional to their slenderness ratio. A red circle holds "safe" branches ($SR_c \leq 70$).

### Usage

```
## S3 method for class 'SR'
plot(x, y = NULL, safeSR = 70, ...)
```

### Arguments

| | |
|---|---|
| `x` | SR object |
| `y` | unused |
| `safeSR` | SR threshold, risky branches are red-coloured |
| `...` | Arguments to be passed to plot.default |

### Value

NULL

### Note

A circle is drawn to encompass the 70- values for slenderness ratio. Branches with 70+ values for the slenderness ratio are considered dangerous. Please note that Mattheck coefficient is corrected to account for branch tilt (the more it deviates from the verticality the higher its coefficient)

### See Also

treeSR

---

## plot.vector — *Plots branches and logs*

---

### Description

Plots branches and logs

The 2d plot represents branches and logs as vectors pointing inwards. Branches to be pruned are not shown on graph.

### Usage

```
## S3 method for class 'vector'
plot(x, y = NULL, txtcol = "grey80", ...)
```

### Arguments

| | |
|---|---|
| x | vectors object |
| y | unused |
| txtcol | Colour of text labels, defaults to "grey80" |
| ... | Arguments to be passed to plot.default |

### Value

NULL

## `plotPolarSegment` *Plots a segment*

### Description

Plots a segmente given two set of polar coordinates (angle, distance from tree base), may be used to represent buildings close to the tree on the CM plot

### Usage

```
plotPolarSegment(a0, d0, a1, d1)
```

### Arguments

| | |
|---|---|
| `a0` | angle of first set of coordinates |
| `d0` | distance of first set of coordinates |
| `a1` | angle of second set of coordinates |
| `d1` | distance of second set of coordinates |

### Value

`NULL`

## `powerEquation` *Returns the result of an exponential equation*

### Description

Returns the result of the exponential equation $Y = a * X^b$ given $a$, $b$ and $X$

### Usage

```
powerEquation(a, b, x)
```

### Arguments

| | |
|---|---|
| `a` | the parameter $a$ in the exponential equation |
| `b` | the parameter $b$ in the exponential equation |
| `x` | the independent variable |

### Value

the dependent variable ($Y$)

### See Also

Other Biomass: allometryABDC; allometryAsca2011; allometryCutini2009; allometryPorte2002; logBiomass; pureQuadraticEquation; treeBiomass; treeTotalBiomass

---

## print.CM *Simple print of Centre of Mass data*

---

### Description

Prints in a human-readable format the cartesian coordinates of tree CM

### Usage

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

### Arguments

| | |
|---|---|
| x | An object of class CM |
| ... | Additional arguments, not used |

### Value

NULL

---

## pureQuadraticEquation *Returns the result of a pure quadratic equation*

---

### Description

Returns the result of the pure quadratic equation $Y = a + bX^2$ given $a$, $b$ and $X$

### Usage

```
pureQuadraticEquation(a, b, x)
```

### Arguments

| | |
|---|---|
| a | the parameter $a$ in the pure quadratic equation |
| b | the parameter $a$ in the pure quadratic equation |
| x | the dependent variable |

### Value

the dependent variable ($Y$)

## See Also

Other Biomass: `allometryABDC`; `allometryAsca2011`; `allometryCutini2009`; `allometryPorte2002`; `logBiomass`; `powerEquation`; `treeBiomass`; `treeTotalBiomass`

---

`setBranchesCM` *Stores branches CM in an object*

---

## Description

Stores branches CM in the object provided as an argument. branchesCM has to be in the range 0.01 - 1

## Usage

```
setBranchesCM(object, value)
```

## Arguments

| | |
|---|---|
| `object` | the object of class `treeData` |
| `value` | the new branch CM |

## Value

the updated list

## Note

Method `treeVectors` must be invoked to take changes into effect

---

`stonePine1FieldData` *Raw CSV file of field recorded values for a stone pine tree*

---

## Description

Required data for the assessment of the centre of mass have been recorded in the field for a stone pine (*Pinus pinea* L.). This is an example of csv file that should be fed to `treeBiomass` to assess tree centre of mass.

### Format

```
 "code","azimuth","dBase","dTip","length","tipD","height","tilt","toBePruned","pathToTip"
"L1",275,73,41,"10.2","2.5",0,80,"FALSE","TRUE" "L2",275,41,16,"3.9","2.75","10.2",80,"FALSE","TRUE"
"B1",190,15,0,"7.95","10.1",,,"FALSE","FALSE" "B2",200,22,0,"7.95","10.4",,,"FALSE","FALSE"
"B3",230,15,0,"7.95","10.4",,,"FALSE","FALSE" "B4",200,18,0,"7.95","11.15",,,"FALSE","FALSE"
"B5",180,7,0,"7.95","11.3",,,"FALSE","FALSE" "B6",150,6,0,"7.95","11.3",,,"FALSE","FALSE"
"B7",340,16,0,"3.95","11.3",,,"FALSE","FALSE" "B8",220,13,0,"7.95","11.8",,,"FALSE","FALSE"
"B9",165,19,0,"7.95","11.8",,,"FALSE","FALSE" "B10",280,8,0,"3.95","11.9",,,"FALSE","FALSE"
"B11",170,9,0,"7.95","11.9",,,"FALSE","FALSE" "B12",265,8,0,"7.95","12.2",,,"FALSE","FALSE"
"B13",75,6,0,"3.95","12.2",,,"FALSE","FALSE" "B14",180,6,0,"7.95","12.2",,,"FALSE","FALSE"
"B15",170,6,0,"7.95","12.6",,,"FALSE","FALSE" "B16",120,5,0,"7.95","12.6",,,"FALSE","FALSE"
"B17",10,14,0,"3.95",13,,,"FALSE","FALSE" "B18",180,13,0,"7.95",13,,,"FALSE","FALSE"
"B19",260,13,0,"7.95","13.2",,,"FALSE","FALSE" "B20",75,6,0,"3.95","13.2",,,"FALSE","FALSE"
"B21",75,10,0,"3.95","13.75",,,"FALSE","FALSE" "B22",215,7,0,"7.95","13.75",,,"FALSE","FALSE"
"B23",140,7,0,"7.95","13.75",,,"FALSE","FALSE" "C",275,16,0,3,3,"14.1",80,"FALSE","TRUE"
```

### Details

This dataset has been collected for a 17.1 metres tall stone pine whose stem was tilted approx. 20 degrees from the vertical plane (or 80 degrees from the horizontal plane). The stem has been sectioned in two logs (`L1` and `L2`), and a final branch (`C`).

The `.csv` file must contain all column headings listed in `importFieldData`, regardless of them being optional (no data in them) or mandatory.

### Source

Original data collected by the author

### Examples

```
library("treecm")
csvFileName <- system.file("data", "stonePine1FieldData.csv.gz", package = "treecm")
treeData <- importFieldData(
  csvFileName,
  650,
  allometryABDC
)
head(treeData$fieldData)
```

---

## stonePine1TreeData *Field recorded values for a stone pine tree*

---

### Description

Required data for the assessment of the centre of mass have been recorded in the field for a stone pine (*Pinus pinea* L.). `treeBiomass` has already been run on the dataset, vectors have yet to be computed.

### Format

The format is: `List of 4 $ fieldData :'data.frame': 26 obs. of 10 variables:`
`..$ azimuth : int [1:26] 275 275 190 200 230 200 180 150 340 220 ...`
`..$ dBase : int [1:26] 73 41 15 22 15 18 7 6 16 13 ... ..$ dTip : num [1:26] 41 16 0 0 0 0 0 0 0 0`
`..$ length : num [1:26] 10.2 3.9 NA NA NA NA NA NA NA NA ... ..$ tipD : num [1:26] 2.5 2.75 7.95 7`
`..$ height : num [1:26] 0 10.2 10.1 10.4 10.4 ... ..$ tilt : num [1:26] 80 80 0 0 0 0 0 0 0 0 ...`
`..$ toBePruned: logi [1:26] FALSE FALSE FALSE FALSE FALSE FALSE ... ..$ pathToTip : logi [1:26] TRUE TR`
`..$ biomass : num [1:26] 1825 193 123 313 123 ... $ density : num 650`
`$ allometryFUN:function (x, diameter) $ branchesCM : num 1`

### Details

This dataset includes a list of 4 elements:

- the `stonePine1FieldData` dataset
- the density of wood
- the allometry function to be used to compute branches biomass
- the estimate of branches centre of mass

### Source

Original data collected by the author

### Examples

```
data(stonePine1TreeData)
vectors  <- treeVectors(stonePine1TreeData)
CM       <- centreOfMass(vectors)
summary(CM)
# The steps to recreate this dataset:
csvFileName <- system.file("data", "stonePine1FieldData.csv.gz", package = "treecm")
treeData <- importFieldData(csvFileName, 650, allometryABDC)
treeData <- treeBiomass(treeData)
```

---

`stonePine2FieldData` *Raw CSV file of field recorded values for a stone pine tree*

---

### Description

Required data for the assessment of the centre of mass have been recorded in the field for a stone pine (*Pinus pinea* L.). This is an example of csv file that should be fed to `treeBiomass` to assess tree centre of mass.

### Format

`"code","azimuth","dBase","dTip","length","tipD","height","tilt","toBePruned","pathToTip"`
`"L1",0,67,40,"6.8",0,0,90,,"TRUE" "B1",250,40,,,"7.8","6.8",,,"B2",240,32,,,"8.9","7.8",,,"B3",55,2`

### Details

This dataset has been collected for a $\approx 11$ metres tall stone pine with a small number of very large branches.

The `.csv` file must contain all column headings listed in `importFieldData`, regardless of them being optional (no data in them) or mandatory.

### Source

Original data collected by the author

### Examples

```
library("treecm")
  treeData <- importFieldData(system.file("data", "stonePine2FieldData.csv.gz", package = "treecm"),
                                   650, allometryABDC)
head(treeData$fieldData)
```

---

## summary.CM *Summary of Centre of Mass data*

---

### Description

Prints in a human-readable format the polar and cartesian coordinates of tree CM

### Usage

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

### Arguments

| | |
|---|---|
| `object` | An object of class `CM` |
| `...` | Additional arguments, not used |

### Value

`NULL`

## toCartesianXY — *Computes the x,y cartesian coordinates*

### Description

Computes the $x$ and $y$ cartesian coordinates from a set of polar coordinates

### Usage

```
toCartesianXY(angle, distance)
```

### Arguments

| | |
|---|---|
| `angle` | The angle in degrees (measured clockwise from the North or any other relevant bearing system defined in the field) |
| `distance` | The distance |

### Value

A vector holding the $x$ and $y$ coordinats expressed in the same unit as the distance argument

### Note

The function assumes the angle is measured clockwise whereas trigonometric functions require a conventional counterclockwise measured angle. Thus the function computes $x$ coordinate as the sine of the angle, and the $y$ coordinate as the cosine of the angle, enabling a correct representation of them on a cartesian plot.

## toCartesianXYZ — *Returns vector cartesian coordinates*

### Description

Given a modulus, a tilt angle and an azimuth angle it returns the vector cartesian coordinates

### Usage

```
toCartesianXYZ(x)
```

### Arguments

| | |
|---|---|
| `x` | a named vector of three elements (z, x, y) |

### Value

a list holding z, x, y coordinates

### See Also

Other Stabilization: anchorRange; centreOfMassAngle; centreOfMassModulus; getPlinthForce; logPathSelection

---

## toPolar — *Converts cartesian (x, y) into polar (angle, distance) coordinates*

---

### Description

Converts cartesian coordinates ($x$, $y$) into polar (angle, distance) ones, assuming (0, 0) as origin of axes and, incidentally, the position of tree base

### Usage

```
toPolar(x, y)
```

### Arguments

| | |
|---|---|
| x | Abscissa coordinate |
| y | Ordinate coordinate |

### Value

A vector holding angle in degrees and distance in the same unit as $x$ and $y$

---

## treeBiomass — *Computes masses of branches and logs*

---

### Description

Computes branches biomass using an allometric function provided in `object$allometryFUN` and logs weight using Smalian's formula.

Branches are telled apart from logs in the raw data frame (`object$fieldData`) because their final diameter is 0 (ie they have a tip) whereas logs have a final diameter > 0.

### Usage

```
treeBiomass(object)
```

### Arguments

| | |
|---|---|
| object | an object of `treeData` class |

### Value

an object of `treeData` class

### See Also

`logBiomass`

Other Biomass: `allometryABDC`; `allometryAsca2011`; `allometryCutini2009`; `allometryPorte2002`; `logBiomass`; `powerEquation`; `pureQuadraticEquation`; `treeTotalBiomass`

---

## treeSR — *Computes slenderness ratio for tree branches*

---

### Description

slenderness ratio is an important index of stability of trees and branches

### Usage

```
treeSR(treeObject, vectorObject)
```

### Arguments

| | |
|---|---|
| `treeObject` | an object of `treeData` class |
| `vectorObject` | an object of `vectors` class |

### Value

an object of class `SR`

### Note

The coefficient takes into account branch angle: $SL_c = \frac{L}{D} \cdot (1 + cos(a))$, where $a$ is the branch angle (0 degrees = horizontal, 90 degrees vertical). Vertical branches have $SL = SL_c$

### References

Mattheck, C. and Breloer, H. *The Body Language of Trees: A Handbook for Failure Analysis (Research for Amenity Trees)* 1995, HMSO (London)

### See Also

`branchSR`

## treeTotalBiomass — *Returns the total biomass of the tree*

### Description

This is just a helper function, it sums the biomass of all logs and branches, as previously computed by `treeBiomass`

### Usage

```
treeTotalBiomass(treeData)
```

### Arguments

| | |
|---|---|
| `treeData` | A named list that includes a `fieldData` data frame element, holding a `biomass`-named column. Note that the `biomass` column is added to the data frame by a previous call to `treeBiomass` function |

### Value

a real number or `FALSE` if the `biomass` column is `NA`

### Note

This function may be used to compute the moment of the tree. Tree biomass (multiplied by standard gravity) is the tree force applied to its CM.

### See Also

Other Biomass: `allometryABDC`; `allometryAsca2011`; `allometryCutini2009`; `allometryPorte2002`; `logBiomass`; `powerEquation`; `pureQuadraticEquation`; `treeBiomass`

### Examples

```
library(treecm)
data(stonePine1TreeData)
print(treeTotalBiomass(stonePine1TreeData))
```

`treeVectors` *Computes cartesian coordinates and moments of branches and logs*

## Description

A data frame is populated with branch and log masses, along with $x$, $y$ cartesian coordinates and $x$, $y$, and $z$ moments. $z$ coordinates and moments are calculated only if branches height from the ground (and tilt) have been measured in the field.

## Usage

```
treeVectors(object)
```

## Arguments

| | |
|---|---|
| `object` | an object of `treeData` class |

## Value

an object of class `vectors`

## See Also

`getCoordinatesAndMoment`

# Index