Description

This package enables you to create dendrograms and tree plots using `ggplot2::ggplot()`. 

Details

The `ggplot2` philosophy is to clearly separate data from the presentation. Unfortunately the plot method for dendrograms (`plot.dendrogram()`) plots directly to a plot device without exposing the data. The `ggdendro` package resolves this by making available functions that extract the dendrogram plot data. This data can be used with `ggplot`. 

The function `dendro_data()` extracts data from different objects that contain dendrogram information. It is a generic function with methods for:

- `hclust`: `dendro_data.hclust()`
- `dendrogram`: `dendro_data.dendrogram()`
- regression trees: `dendro_data.tree()`
- partition trees: `dendro_data.rpart()`
- agnes and diana: `dendro_data.twins()`

These methods create an object of class `dendro`, consisting of a list of data frames. To extract the relevant data frames from the list, you can use the accessor functions:

- `segment()`: the line segment data
- `label()`: the text for each end segment
- `leaf_label()`: the leaf labels of a tree diagram

To plot a dendrogram, either construct a plot with `ggplot2::ggplot()` or use the function `ggdendrogram()`.

Author(s)

Andrie de Vries - <apdevries@gmail.com>
as.dendro

See Also
dendro_data()

dendro_data

Coerces object to class dendro.

Description
Method for coercing object to class dendro.

Usage
as.dendro(segments, labels, leaf_labels = NULL, class)

Arguments
segments data.frame with segment data
labels data.frame with labels data
leaf_labels data.frame with leaf label data
class The class of the original model object, e.g. "hclust". This is used by ggdendrogram() to determine the angle and justification of labels

See Also
dendro_data() and ggdendro-package()

dendro_data

Extract cluster data from a model into a list of data frames.

Description
This function provides a generic mechanism to extract relevant plotting data, typically line segments and labels, from a variety of cluster models.

Extract line segment and label data from stats::dendrogram() or stats::hclust() object. The resulting object is a list of data frames containing line segment data and label data.
Usage

dendro_data(model, ...)

## Default S3 method:
dendro_data(model, ...)

## S3 method for class 'dendrogram'
dendro_data(model, type = c("rectangle", "triangle"), ...)

## S3 method for class 'hclust'
dendro_data(model, type = c("rectangle", "triangle"), ...)

## S3 method for class 'twins'
dendro_data(model, type = c("rectangle", "triangle"), ...)

Arguments

model object of type \texttt{stats::hclust()}, \texttt{stats::dendrogram()} or \texttt{tree::tree()}

... ignored

type The type of plot, indicating the shape of the dendrogram. "rectangle" will draw rectangular lines, while "triangle" will draw triangular lines.

Details

For \texttt{stats::dendrogram()} and \texttt{tree::tree()} models, extracts line segment data and labels.

Value

a list of data frames that contain the data appropriate to each cluster model

A list with components:

segments Line segment data

labels Label data

See Also

There are several implementations for specific cluster algorithms:

- \texttt{dendro_data.hclust()}
- \texttt{dendro_data.dendrogram()}
- \texttt{dendro_data.tree()}
- \texttt{dendro_data.rpart()}

To extract the data for line segments, labels or leaf labels use:

- \texttt{segment()}: the line segment data
- \texttt{label()}: the text for each end segment
• **leaf_label()**: the leaf labels of a tree diagram

`ggdendrogram()`

Other dendro_data methods: `dendro_data.rpart()`, `dendro_data.tree()`, `dendrogram_data()`, `rpart_labels()`

Other dendrogram/hclust functions: `dendrogram_data()`

### Examples

```r
require(ggplot2)

### Demonstrate dendro_data.dendrogram

model <- hclust(dist(USArrests), "ave")
dendro <- as.dendrogram(model)

# Rectangular lines
ddata <- dendro_data(dendro, type = "rectangle")
ggplot(segment(ddata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  coord_flip() +
  scale_y_reverse(expand = c(0.2, 0)) +
  theme_dendro()

# Triangular lines
ddata <- dendro_data(dendro, type = "triangle")
ggplot(segment(ddata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_dendro()

### Demonstrate dendro_data.hclust

require(ggplot2)

hc <- hclust(dist(USArrests), "ave")

# Rectangular lines
hcdata <- dendro_data(hc, type = "rectangle")
ggplot(segment(hcdata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  coord_flip() +
  scale_y_reverse(expand = c(0.2, 0)) +
  theme_dendro()

# Triangular lines
hcdata <- dendro_data(hc, type = "triangle")
ggplot(segment(hcdata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_dendro()

### Demonstrate the twins of agnes and diana, from package cluster

if (require(cluster)) {
  model <- agnes(votes.repub, metric = "manhattan", stand = TRUE)
  ...}
```
dendro_data.rpart

Extract data from classification tree object for plotting using ggplot.

Description

Extracts data to plot line segments and labels from a `rpart::rpart()` classification tree object. This data can then be manipulated or plotted, e.g. using `ggplot2::ggplot()`.

Usage

```r
## S3 method for class 'rpart'
dendro_data(
  model, 
  uniform = FALSE, 
  branch = 1, 
  compress = FALSE, 
  nspace, 
  minbranch = 0.3, 
  ...
)
```

Arguments

- **model**: object of class "tree", e.g. the output of `tree()`
- **uniform**: if TRUE, uniform vertical spacing of the nodes is used; this may be less cluttered when fitting a large plot onto a page. The default is to use a non-uniform spacing proportional to the error in the fit.
- **branch**: controls the shape of the branches from parent to child node. Any number from 0 to 1 is allowed. A value of 1 gives square shouldered branches, a value of 0 give V shaped branches, with other values being intermediate.
- **compress**: if FALSE, the leaf nodes will be at the horizontal plot coordinates of 1:nleaves. If TRUE, the routine attempts a more compact arrangement of the tree. The compaction algorithm assumes uniform=TRUE; surprisingly, the result is usually an improvement even when that is not the case.


*dendro_data.rpart*

```r

dendro_data.rpart

nspace

the amount of extra space between a node with children and a leaf, as compared
to the minimal space between leaves. Applies to compressed trees only. The
default is the value of branch.

minbranch

set the minimum length for a branch to minbranch times the average branch
length. This parameter is ignored if uniform=TRUE. Sometimes a split will give
very little improvement, or even (in the classification case) no improvement at
all. A tree with branch lengths strictly proportional to improvement leaves no
room to squeeze in node labels.

... ignored

Details

This code is in essence a copy of `rpart::plot.rpart()`, retaining the plot data but without plotting
to a plot device.

Value

A list of three data frames:

- `segments` a data frame containing the line segment data
- `labels` a data frame containing the label text data
- `leaf_labels` a data frame containing the leaf label text data

See Also

- `ggdendrogram()`

Other dendro_data methods: `dendro_data.tree()`, `dendro_data()`, `dendrogram_data()`, `rpart_labels()`

Other rpart functions: `rpart_labels()`, `rpart_segments()`

Examples

```r

### Demonstrate rpart

if (require(rpart)) {
  require(ggplot2)
  fit <- rpart(Kyphosis ~ Age + Number + Start, method = "class",
               data = kyphosis)
  fitr <- dendro_data(fit)
  ggplot() +
    geom_segment(data = fitr$segments,
                 aes(x = x, y = y, xend = xend, yend = yend)) +
    geom_text(data = fitr$labels, aes(x = x, y = y, label = label)) +
    geom_text(data = fitr$leaf_labels, aes(x = x, y = y, label = label)) +
    theme_dendro()
}
```
dendro_data.tree

Extract data from regression tree object for plotting using ggplot.

Description

Extracts data to plot line segments and labels from a tree::tree() object. This data can then be manipulated or plotted, e.g. using ggplot2::ggplot().

Usage

## S3 method for class 'tree'
dendro_data(model, type = c("proportional", "uniform"), ...)

Arguments

model  
object of class "tree", e.g. the output of tree()

type  
Either proportional or uniform. If this partially matches "uniform", the branches are of uniform length. Otherwise they are proportional to the decrease in impurity.

...  
ignored

Value

A list of three data frames:

segments  
a data frame containing the line segment data

labels  
a data frame containing the label text data

leaf_labels  
a data frame containing the leaf label text data

Author(s)

Andrie de Vries, using code modified from original by Brian Ripley

See Also

ggdendrogram()

Other dendro_data methods: dendro_data.rpart(), dendro_data(), dendrogram_data(), rpart_labels()

Other tree functions: get_data_tree_leaf_labels(), tree_labels(), tree_segments()

Examples

### Demonstrate tree

if (require(tree)) {
  require(ggplot2)
  require(MASS)
  data(cpus, package = "MASS")
```r
cpus.ltr <- tree(log10(perf) ~ syct + mmin + mmax + cach + chmin + chmax,
    data = cpus)
tree_data <- dendro_data(cpus.ltr)
ggplot(segment(tree_data)) +
    geom_segment(aes(x = x, y = y, xend = xend, yend = yend, size = n),
        colour = "lightblue"
    ) +
    scale_size("n") +
    geom_text(
        data = label(tree_data),
        aes(x = x, y = y, label = label), vjust = -0.5, size = 4
    ) +
    geom_text(
        data = leaf_label(tree_data),
        aes(x = x, y = y, label = label), vjust = 0.5, size = 3
    ) +
    theme_dendro()
}
```

---

**ggdendrogram**

*Creates dendrogram plot using ggplot.*

**Description**

This is a convenience function

**Usage**

```r
ggdendrogram(
    data,
    segments = TRUE,
    labels = TRUE,
    leaf_labels = TRUE,
    rotate = FALSE,
    theme_dendro = TRUE,
    ...
)
```

**Arguments**

- **data**: Either a dendro object or an object that can be coerced to class dendro using the `dendro_data()` function, i.e. objects of class dendrogram, hclust or tree
- **segments**: If TRUE, show line segments
- **labels**: if TRUE, shows segment labels
- **leaf_labels**: if TRUE, shows leaf labels
- **rotate**: if TRUE, rotates plot by 90 degrees
- **theme_dendro**: if TRUE, applies a blank theme to plot (see `theme_dendro()`)  
- **...**: other parameters passed to `ggplot2::geom_text()`
## is.dendro

Tests whether an object is of class dendro.

### Description

Is a dendro? Tests whether an object is of class dendro.

### Usage

```r
is.dendro(x)
```

### Arguments

- `x` Object to check

### See Also

`dendro_data()` and `ggdendro-package()`

---

### Examples

#### Demonstrate ggdendrogram

```r
library(ggplot2)
hc <- hclust(dist(USArrests), "ave")

# Demonstrate plotting directly from object class hclust
p <- ggdendrogram(hc, rotate = FALSE)
print(p)

# demonstrate converting hclust to dendro using dendro_data first
hcdata <- dendro_data(hc)
ggdendrogram(hcdata, rotate = TRUE, size = 2) +
  labs(title = "Dendrogram in ggplot2")
```
segment

Returns segment, label or leaf-label data from dendro object.

Description

segment extracts line segments, label extracts labels, and leaf_label extracts leaf labels from a dendro object.

Usage

segment(x)

label(x)

leaf_label(x)

Arguments

x dendro object

See Also

dendro_data()

theme_dendro

Creates completely blank theme in ggplot.

Description

Sets most of the ggplot options to blank, by returning blank theme elements for the panel grid, panel background, axis title, axis text, axis line and axis ticks.

Usage

theme_dendro()
## Index

<table>
<thead>
<tr>
<th><strong>dendro_data methods</strong></th>
<th>plot.dendrogram(), 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dendro_data, 3</td>
<td>rpart::plot.rpart(), 7</td>
</tr>
<tr>
<td>dendro_data.rpart, 6</td>
<td>rpart::rpart(), 6</td>
</tr>
<tr>
<td>dendro_data.tree, 8</td>
<td>rpart_labels, 5, 7, 8</td>
</tr>
<tr>
<td><strong>dendrogram/hclust functions</strong></td>
<td>rpart_segments, 7</td>
</tr>
<tr>
<td>dendro_data, 3</td>
<td>segment, 11</td>
</tr>
<tr>
<td><strong>package</strong></td>
<td>segment(), 2, 4</td>
</tr>
<tr>
<td>dendro_data.rpart, 6</td>
<td>stats::dendrogram(), 3, 4</td>
</tr>
<tr>
<td><strong>rpart functions</strong></td>
<td>stats::hclust(), 3, 4</td>
</tr>
<tr>
<td>dendro_data.tree, 8</td>
<td>theme_dendro, 11</td>
</tr>
<tr>
<td><strong>tree functions</strong></td>
<td>theme_dendro(), 9</td>
</tr>
<tr>
<td>dendro_data, 3</td>
<td>tree_labels, 8</td>
</tr>
<tr>
<td>as.dendro, 3</td>
<td>tree_segments, 8</td>
</tr>
<tr>
<td>dendro_data, 3, 7, 8</td>
<td>dendro_data(), 2, 3, 9–11</td>
</tr>
<tr>
<td>dendro_data(), 2, 3, 9–11</td>
<td>dendro_data.dendrogram(), 2, 4</td>
</tr>
<tr>
<td>dendro_data(), 2, 4</td>
<td>dendro_data.hclust(), 2, 4</td>
</tr>
<tr>
<td>dendro_data.rpart, 5, 6, 8</td>
<td>dendro_data.rpart(), 2, 4</td>
</tr>
<tr>
<td>dendro_data.tree, 5, 7, 8</td>
<td>dendro_data.tree(), 2, 4</td>
</tr>
<tr>
<td>dendro_data.tree(), 2, 4</td>
<td>dendro_data.twins(), 2</td>
</tr>
<tr>
<td>dendrogram_data, 5, 7, 8</td>
<td>get_data_tree_leaf_labels, 8</td>
</tr>
<tr>
<td>ggdendro (ggdendro-package), 2</td>
<td>ggdendro-package, 2</td>
</tr>
<tr>
<td>ggdendrogram, 9</td>
<td>ggdendrogram(), 2, 3, 5, 7, 8</td>
</tr>
<tr>
<td>ggdendrogram(), 2, 3, 5, 7, 8</td>
<td>ggplot2::geom_text(), 9</td>
</tr>
<tr>
<td>ggplot2::ggplot(), 2, 6, 8, 10</td>
<td>is.dendro, 10</td>
</tr>
<tr>
<td><strong>leaf functions</strong></td>
<td>label(), 2, 4</td>
</tr>
<tr>
<td>label(), 2, 4</td>
<td>leaf_label(), 2, 5</td>
</tr>
<tr>
<td>leaf_label(), 2, 5</td>
<td>leaf_label(), 2, 5</td>
</tr>
<tr>
<td>leaf_label(segment), 11</td>
<td>leaf_label(), 2, 5</td>
</tr>
</tbody>
</table>