Using the ggdendro package for plotting dendrograms and tree diagrams

Andrie de Vries

November 23, 2012

ggdendro is a package that makes it easy to extract dendrogram and tree diagrams into a data frame.

1 Introduction

The ggdendro package provides a general framework to extract the plot data for a dendrograms and tree diagrams.

It does this by providing generic function dendro_data that will extract the appropriate segment data as well as labels. This data is returned as a list of data.frames. These data frames can be extracted using three accessor functions:

- segment
- label
- leaf_label

The package also provides two convenient wrapper functions:

- ggdendrogram is a wrapper around ggplot to create a dendrogram using a single line of code. The resulting object is of class ggplot, so can be manipulated using the ggplot tools.
- theme_dendro is a ggplot theme with a blank canvas, i.e. no axes, axis labels or tick marks.

The ggplot package doesn't get loaded automatically, so remember to load it first:

```
> library(ggplot2)
> library(ggdendro)
```

2 Using the ggdendrogram wrapper

The ggdendro package will extract all of the plot data from dendrogram objects. Sometimes it is useful to have fine-grained control over the plot. Other times it might be more convenient to have a simple wrapper around ggplot to produce a dendrogram with a small amount of code.

The function ggdendrogram provides such a wrapper to produce a plot with a single line of code. It provides a few options for controlling the display of line segments, labels and plot rotation (rotated by 90 degrees or not).

```
> hc <- hclust(dist(USArrests), "ave")
> p <- ggdendrogram(hc, rotate=FALSE, size=2)
> print(p)
```

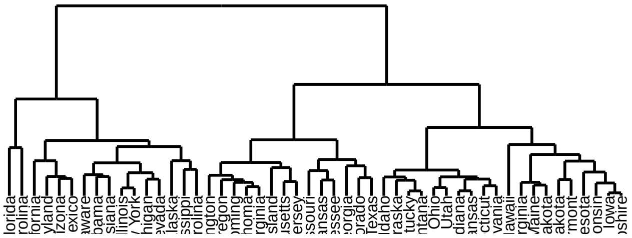

Figure 1: A dendrogram produced using ggdendrogram

The next section shows how to take full control over the data extraction and subsequent plotting.

3 Extracting the dendrogram plot data using dendro_data

The hclust and dendrogram functions in R makes it easy to plot the results of hierarchical cluster analysis and other dendrograms in R. However, it is hard to extract the data from this analysis to customise these plots, since the plot functions for both these classes prints directly without the option of returning the plot data.

```
> hc <- hclust(dist(USArrests), "ave")
> dhc <- as.dendrogram(hc)
> # Rectangular lines
> ddata <- dendro_data(dhc, type="rectangle")
> p <- ggplot(segment(ddata)) +
+     geom_segment(aes(x=x, y=y, xend=xend, yend=yend)) +
+     coord_flip() + scale_y_reverse(expand=c(0.2, 0))
> print(p)
```

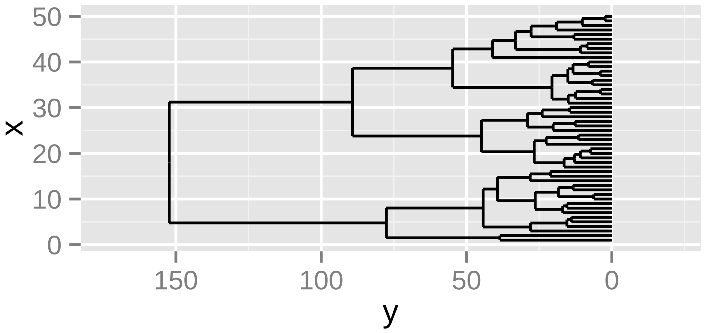

Figure 2: A dendrogram produced using `dendro_data` and `ggplot`

Of course, using ggplot to create the dendrogram means one has full control over the appearance of the plot. For example, here is the same data, but this time plotted horizontally with a clean background. In `ggplot` this means passing a number of options to `theme`. `ggdendro` has a convenient function, `theme_dendro` that wraps these options into a convenient function.

```
> p <- p + coord_flip() + theme_dendro()
> print(p)
```

Dendrograms can also be drawn using triangular lines instead of rectangular lines. For example:

```
> ddata <- dendro_data(dhc, type="triangle")
> p <- 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()
> print(p)
```

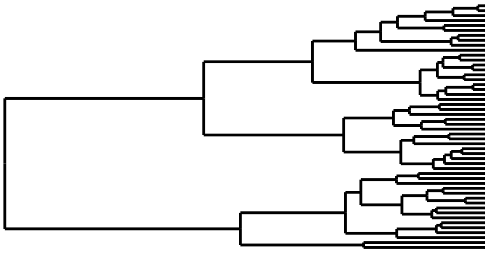

Figure 3: Dendrogram rotated on clear background

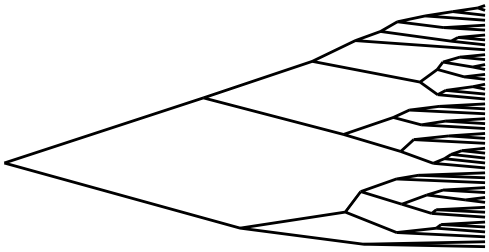

Figure 4: A dendrogram with triangular connection lines

4 Regression tree diagrams

The `tree` function in package `tree` creates tree diagrams. To extract the plot data for these diagrams using `ggdendro` follows the same basic pattern as dendrograms:

```
> require(tree)
> data(cpus, package="MASS")
> cpus.ltr <- tree(log10(perf) ~
+         syct+mmin+mmax+cach+chmin+chmax, cpus)
> tree_data <- dendro_data(cpus.ltr)
> p <- ggplot(segment(tree_data)) +
+     geom_segment(aes(x=x, y=y, xend=xend, yend=yend, size=n),
+         colour="blue", alpha=0.5) +
+     scale_size("n") +
+     geom_text(data=label(tree_data),
+         aes(x=x, y=y, label=label), vjust=-0.5, size=3) +
+     geom_text(data=leaf_label(tree_data),
+         aes(x=x, y=y, label=label), vjust=0.5, size=2) +
+     theme_dendro()
> print(p)
```

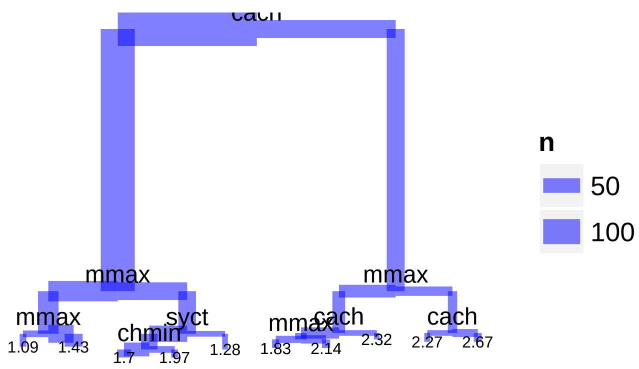

Figure 5: Regression tree plot

5 Classification tree diagrams

The `rpart` function in package `rpart` creates classification diagrams. To extract the plot data for these diagrams using `ggdendro` follows the same basic pattern as dendrograms:

```
> library(rpart)
> fit <- rpart(Kyphosis ˜ Age + Number + Start,
+     method="class", data=kyphosis)
> fitr <- dendro_data(fit)
> p <- 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), size=3, vjust=0) +
+     geom_text(data=fitr$leaf_labels,
+         aes(x=x, y=y, label=label), size=3, vjust=1) +
+     theme_dendro()
> print(p)
```

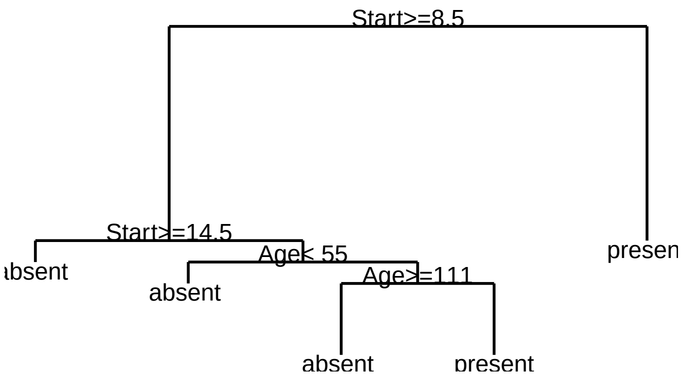

Figure 6: Classification tree plot

6 Conclusion

The `ggdendro` package makes it easy to extract the line segment and label data from hclust, dendrogram and tree objects.