Introduction to the ggcluster package for plotting cluster analysis results

Andrie de Vries

July 13, 2011

ggcluster is a package that makes it easy to extract results of cluster analysis into a data frame. It provides a common framework for a number of hierarchical and other cluster analysis techniques, including:

- hclust() - hierarchical clustering
- dendrogram()- dendrograms
- kmeans() - k-means clustering
- Mclust() - model based clustering

1 Introduction

Most of the cluster analysis functions in R, e.g. `hclust()` and `dendrogram()` provide `plot()` functions to display the results. However, it is sometimes hard to extract the data from this analysis to customise these plots, especially when the `plot()` functions don't provide an option to return the data.

The package provides a general framework to extract the plot data for a variety of cluster analysis functions.

It does this by providing generic function `cluster_data` that will extract the appropriate cluster data as well as labels. This data is returned as a list of data.frames. These data frames can

```
> library(ggplot2)
> library(ggcluster)
```

2 Dendrograms

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)
> ddata <- cluster_data(dhc, type = "rectangle")
> p <- ggplot(ddata$segments) + geom_segment(aes(x = x0, y = y0,
+     xend = x1, yend = y1)) + coord_flip() + scale_y_reverse(expand = c(0.2,
+     0))
> print(p)
```

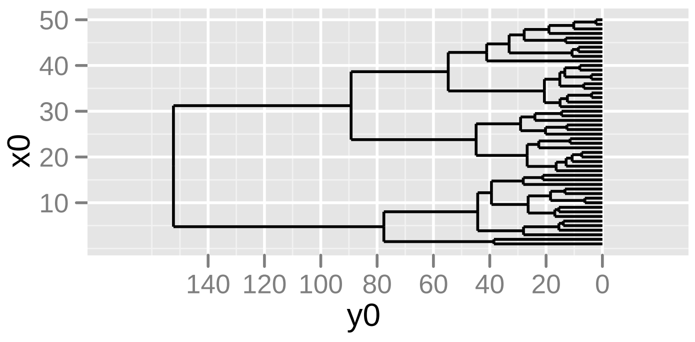

Figure 1: A dendrogram produced using `cluster_data` and `ggplot()`

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:

```
> p <- p + coord_flip() + opts(panel.grid.major = theme_blank(),
+     panel.grid.minor = theme_blank(), panel.background = theme_blank(),
+     axis.title.x = theme_text(colour = NA), axis.title.y = theme_blank(),
+     axis.text.x = theme_blank(), axis.text.y = theme_blank(),
+     axis.line = theme_blank())
> print(p)
```

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

```
> ddata <- cluster_data(dhc, type = "triangle")
> p <- ggplot(ddata$segments) + geom_segment(aes(x = x0, y = y0,
+     xend = x1, yend = y1)) + coord_flip() + scale_y_reverse(expand = c(0.2,
+     0)) + opts(panel.grid.major = theme_blank(), panel.grid.minor = theme_blank(),
+     panel.background = theme_blank(), axis.title.x = theme_text(colour = NA),
+     axis.title.y = theme_blank(), axis.text.x = theme_blank(),
+     axis.text.y = theme_blank(), axis.line = theme_blank())
> print(p)
```

3 Tree diagrams

`ggcluster` can also deal with tree plots.

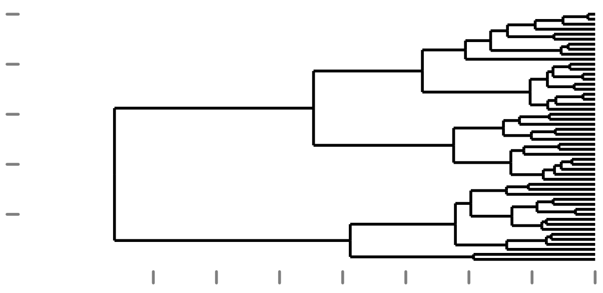

Figure 2: Dendrogram rotated on clear background

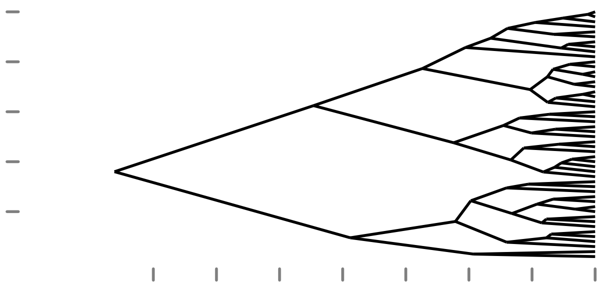

Figure 3: A dendrogram with triangular connection lines

```
> data(cpus, package = "MASS")
> cpus.ltr <- tree(log10(perf) ~ syct + mmin + mmax + cach + chmin +
+     chmax, cpus)
> tree_data <- cluster_data(cpus.ltr)
> p <- ggplot(tree_data$segments) + geom_segment(aes(x = x, y = y,
+     xend = xend, yend = yend, size = n), colour = "blue", alpha = 0.5) +
+     scale_size("n", to = c(0, 3)) + geom_text(data = tree_data$labels,
+     aes(x = x, y = y, label = label), vjust = -0.5, size = 4) +
+     geom_text(data = tree_data$leaf_labels, aes(x = x, y = y,
+         label = label), vjust = 0.5, size = 3)
> print(p)
```

4 Cluster analysis

Some text here.

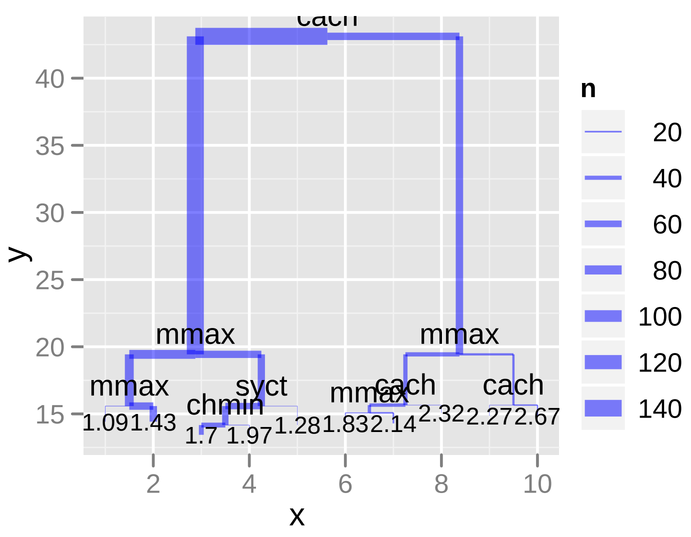

Figure 4: Tree plot

5 Final thoughts

The last word.