cluster jerarquico en R

Jaime Isaac

22/4/2021

cluster jerarquico en R Ejemplo 1

Introducción

Aprenderá la visualización mejorada del dendrograma de agrupamiento con R studio. Este tema cubrirá los siguientes aspectos:

- Cálculo de la matriz de distancia
- Agrupación jerárquica
- Personalización de dendrograma

Datos de importacion

El archivo de datos utilizado aquí se obtiene del conjunto de datos de demostración de R **USArrests.** El uso de la función `head()`imprimirá las primeras seis filas del conjunto de datos `USArrest`s. La función`str()` muestra la estructura interna del conjunto de datos.

```
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

```
str(USArrests)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

Conjunto de datos escalado

Para escalar los datos, utilice la función scale(). scale es una función genérica cuyo método predeterminado centra y / o escala las columnas de una matriz numérica. Si el argumento center se establece en VERDADERO, el centrado se realiza tomando las desviaciones medias de cada columna. Si scale = TRUE, la escala se realiza dividiendo las columnas de datos (centradas) por sus desviaciones estándar.

```
data.scaled <- scale(x = USArrests,
                     center = TRUE,
                     scale = TRUE)
head(data.scaled)
```

```
##                Murder   Assault   UrbanPop         Rape
## Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
## Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
## Arizona    0.07163341 1.4788032  0.9989801  1.042878388
## Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
## California 0.27826823 1.2628144  1.7589234  2.067820292
## Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```

Cálculo de la matriz de distancia.

La matriz de distancia se puede calcular usando la función dist (). Esta función calcula y devuelve la matriz de distancia calculada utilizando la medida de distancia especificada para calcular las distancias entre las filas de una matriz de datos. Para ejecutar esta función, asegúrese de que el paquete de estadísticas esté cargado usando la función require ().

El argumento x especifica una matriz numérica, un marco de datos o un objeto "dist". El método del segundo argumento especifica la medida de distancia que se utilizará. El método debe ser uno de los siguientes:

"euclidiana", "máximo", "manhattan", "canberra", "binario" o "minkowski"

Calcule los resultados del objeto res.dist como una matriz utilizando la función as.matrix () y especifique el número de filas y columnas que se imprimirán entre corchetes.

```
require(stats)
# Distance matrix computation
res.dist = dist(x = data.scaled,
                method = "euclidean")
# Print distance matrix
output = as.matrix(res.dist)[1:6, 1:6]
round(output, digits = 3)
```

```
##            Alabama Alaska Arizona Arkansas California Colorado
## Alabama      0.000  2.704   2.294    1.290      3.263    2.651
## Alaska       2.704  0.000   2.701    2.826      3.013    2.327
## Arizona      2.294  2.701   0.000    2.718      1.310    1.365
## Arkansas     1.290  2.826   2.718    0.000      3.764    2.831
## California   3.263  3.013   1.310    3.764      0.000    1.288
## Colorado     2.651  2.327   1.365    2.831      1.288    0.000
```

El método "dist" de as.matrix() y as.dist() se puede utilizar para la conversión entre objetos de la clase "dist" y matrices de distancia convencionales.

```
d = as.dist(output)
d
```

```
##             Alabama   Alaska  Arizona Arkansas California
## Alaska     2.703754
## Arizona    2.293520 2.700643
## Arkansas   1.289810 2.826039 2.717758
## California 3.263110 3.012541 1.310484 3.763641
## Colorado   2.651067 2.326519 1.365031 2.831051   1.287619
```

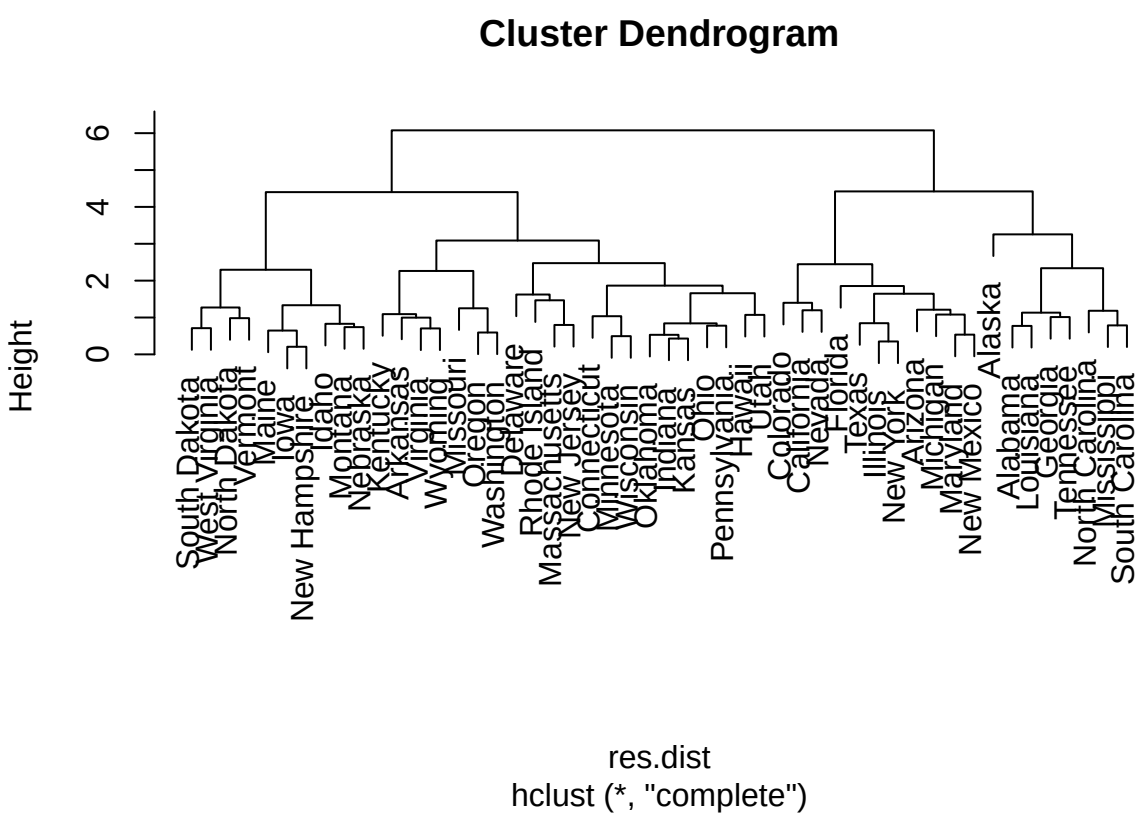
Agrupación jerárquica

El agrupamiento jerárquico es un análisis de agrupamiento sobre un conjunto de diferencias y métodos para analizarlo. Dicho agrupamiento se realiza mediante el uso de la hclust()función en el paquete de estadísticas .

El argumento despecifica una estructura de disimilitud producida por la dist()función. El segundo argumento es el methodque especifica el método de aglomeración que se utilizará. Debe ser uno de los siguientes:

“ward.D”, “ward.D2”, “single”, “complete”, “average” (UPGMA), “mcquitty” (WPGMA), “median” (WPGMC) or “centroid” (UPGMC).

```
# Cluster dendrogram using stats package
require(stats)
res.hc <- hclust(d = res.dist,
                 method = "complete")
plot(x = res.hc)
```



Otra forma de visualización mejorada del dendrograma es mediante el uso de factoextrapackage. La función fviz_dend()dibuja fácilmente dendrogramas hermosos utilizando la plot()función o ggplot2()función base R. También proporciona una opción para dibujar un dendrograma circular y árboles filogénicos.

El x argumento especifica un objeto de la clase dendrogram, hclust, agnes, diana, hcut, hkmeans o HCPC. El tamaño de las etiquetas y el ancho de la línea del rectángulo se pueden controlar estableciendo el valor de cex y lwd como argumentos.

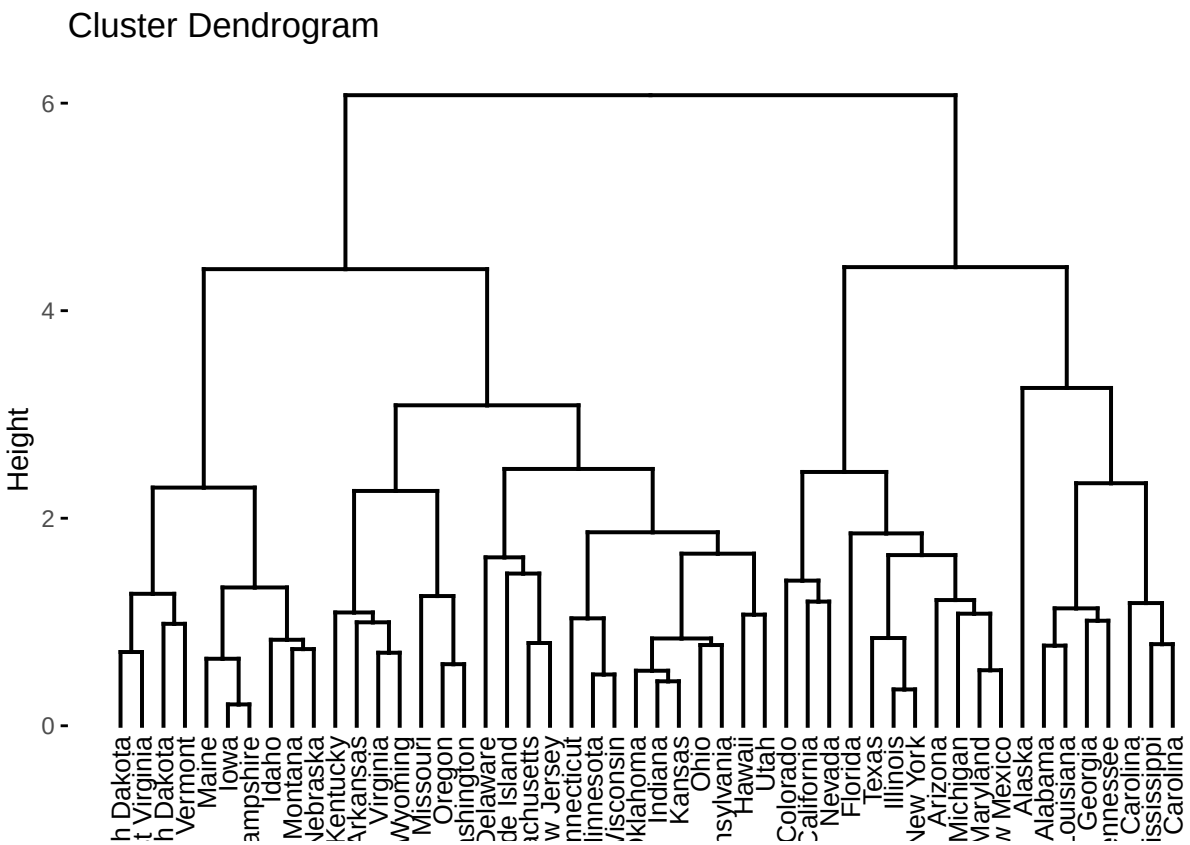
```
# Cluster dendrogram using factoextra package
require(factoextra)
```

```
## Loading required package: factoextra
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_dend(x = res.hc, cex = 0.7, lwd = 0.7)
```



Personalizar dendrograma

Opciones de color Los grupos en el dendrograma se pueden asignar con diferentes nombres de color integrados en R. La colors()función devuelve los nombres de colores incorporados que R conoce.

require(grDevices) colors()

Las paletas de gráficos se pueden configurar o visualizar mediante la palette() función. En R, casi siempre es mejor especificar los colores por su nombre. La forma rápida de mostrar colores en un gráfico es mediante la show_col() función.

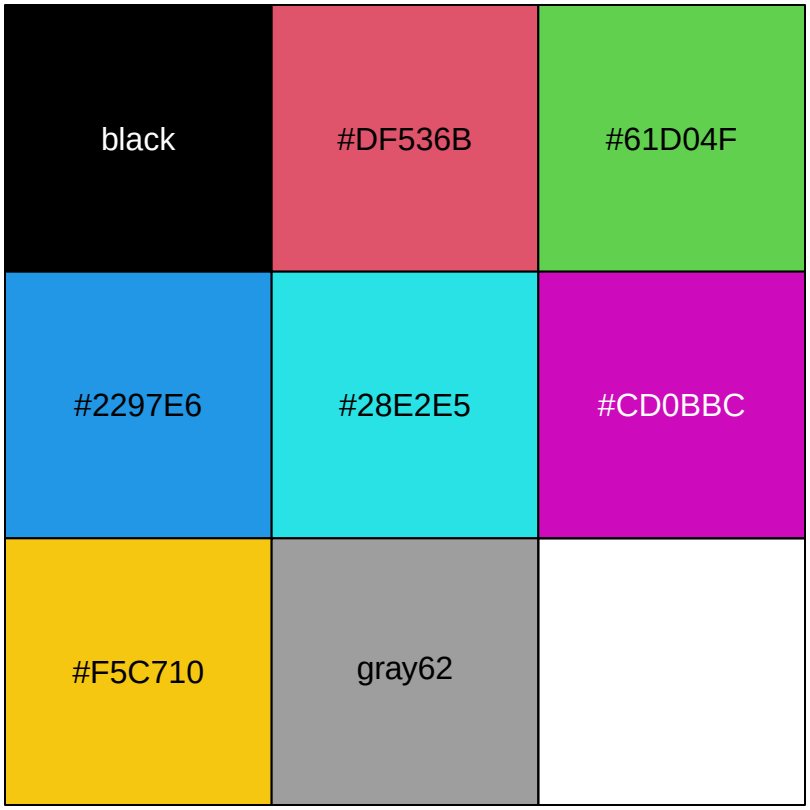
```
require(scales)
```

```
## Loading required package: scales
```

```
palette()
```

```
## [1] "black"   "#DF536B" "#61D04F" "#2297E6" "#28E2E5" "#CD0BBC" "#F5C710"
## [8] "gray62"
```

```
show_col(palette(rainbow(6)))
```

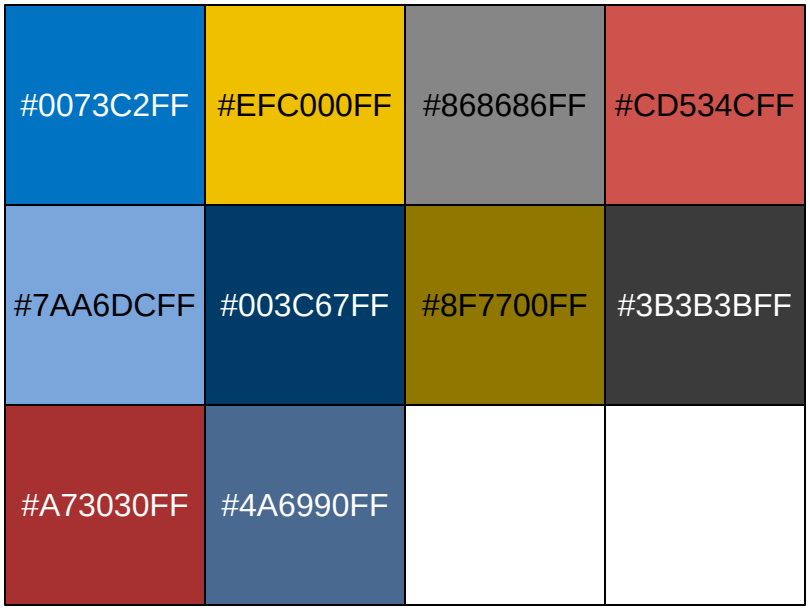


Para ver las paletas de colores inspiradas en gráficos en la pal_jco()función de uso de revista de oncología clínica . El argumento paletteespecifica el tipo de paleta. Actualmente hay una opción disponible default(paleta de 10 colores). El argumento alfa especifica el nivel de transparencia. El valor de este argumento puede estar entre [Error de procesamiento matemático]y [Error de procesamiento matemático].

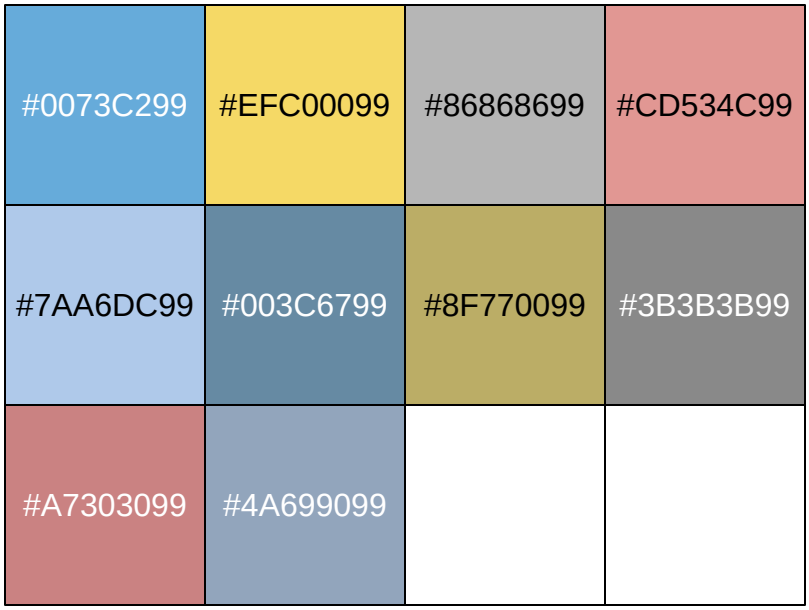
```
require("ggsci")
```

```
## Loading required package: ggsci
```

```
show_col(pal_jco(palette = c("default"))(10))
```




```
show_col(pal_jco("default", alpha = 0.6)(10))
```



Asignar colores y dibujar rectángulos Se pueden agregar colores para el número de grupos o clústeres tanto para líneas como para rectángulos. El argumento k_colorsespecifica un vector que contiene colores que se utilizarán para cada grupo. Los valores permitidos también incluyen "gris" para las paletas de colores grises; paletas de cerveza y paletas de revistas científicas del paquete ggsci R.

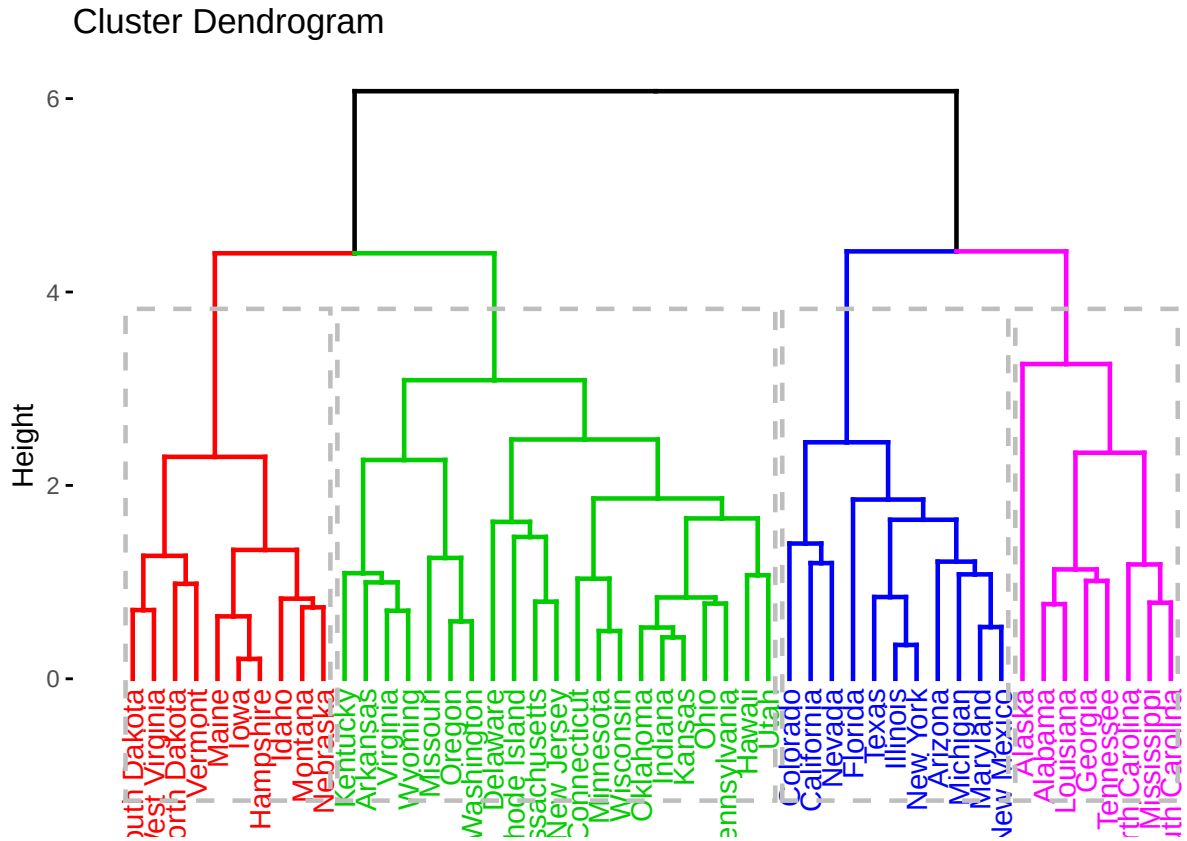
ggsci: "Npg", "aaas", "lancet", "jco", "ucscgb", "uchicago", "simpsons" y "rickandmorty"

El argumento rectespecifica un valor lógico que indica si se debe agregar un rectángulo alrededor de los grupos.

Used only when k != NULL.

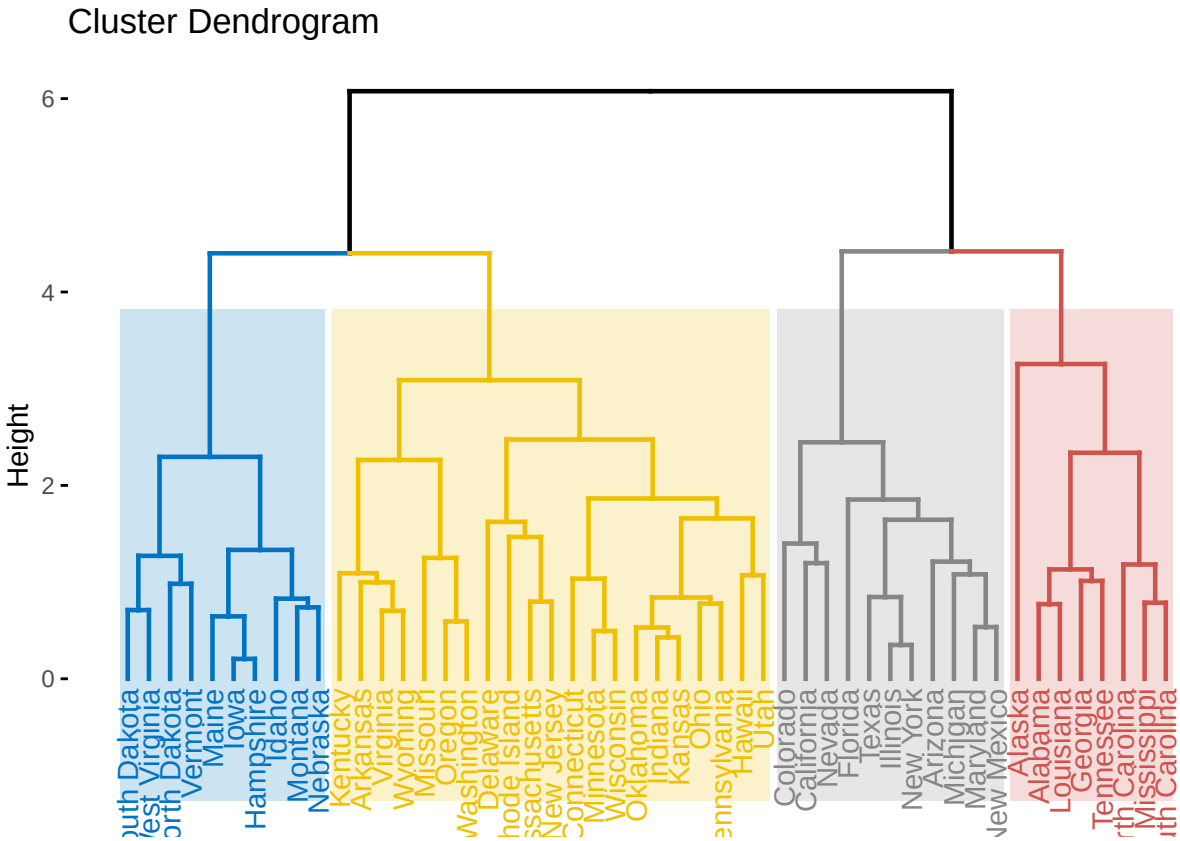
El color del borde y el tipo de línea de los rectángulos se pueden personalizar mediante el uso de rect_border argumentos. El rect_filles un argumento lógico si es VERDADERO, llene el rectángulo.

```
fviz_dend(x = res.hc, cex = 0.8, lwd = 0.8, k = 4,
# Manually selected colors
          k_colors = c("red", "green3", "blue", "magenta"),
          rect = TRUE,
          rect_border = "gray",
          rect_fill = FALSE)
```



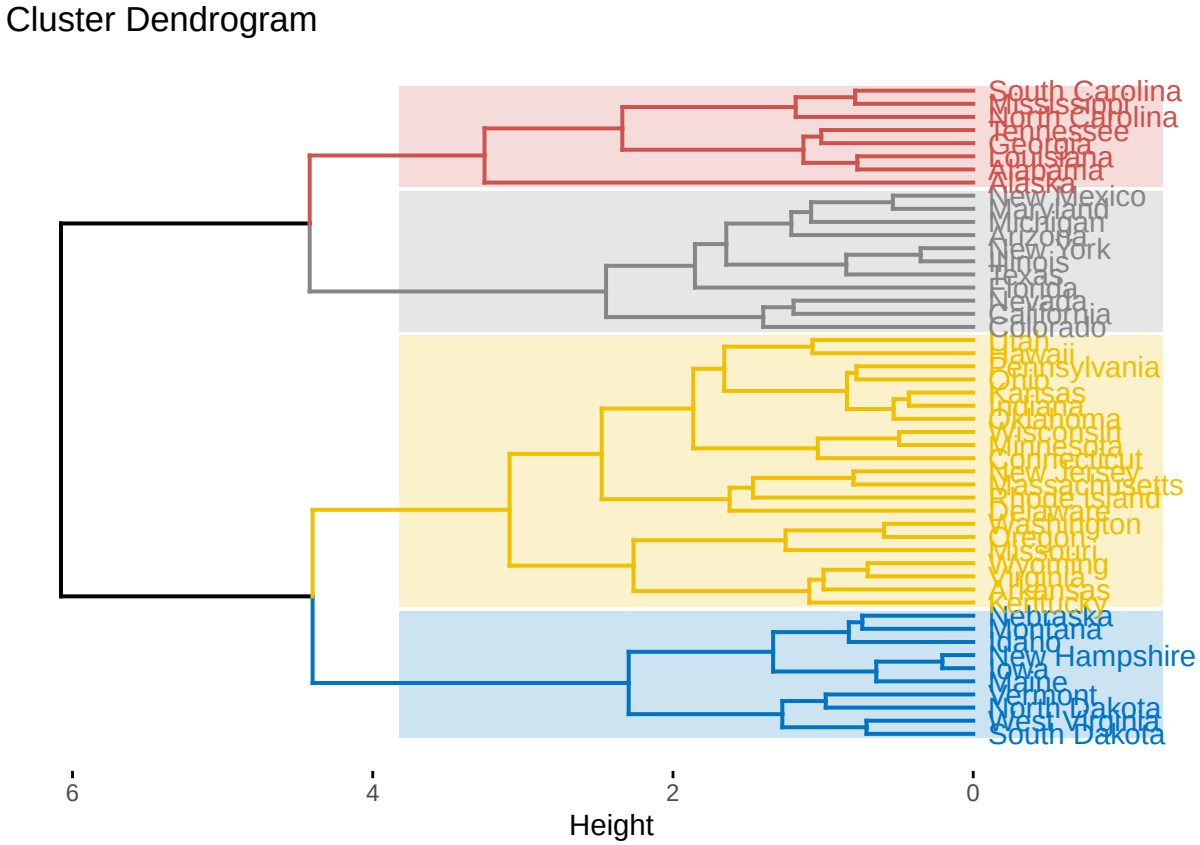

```
fviz_dend(x = res.hc, cex = 0.8, lwd = 0.8, k = 4,


# OR JCO fill color for rectangles
          k_colors = c("jco"),
          rect = TRUE,
          rect_border = "jco",
          rect_fill = TRUE)
```



Alineación horizontal La alineación del dendrograma se puede cambiar estableciendo un valor lógico para el horizargumento. La configuración TRUEde este argumento dibujará un dendrograma horizontal.

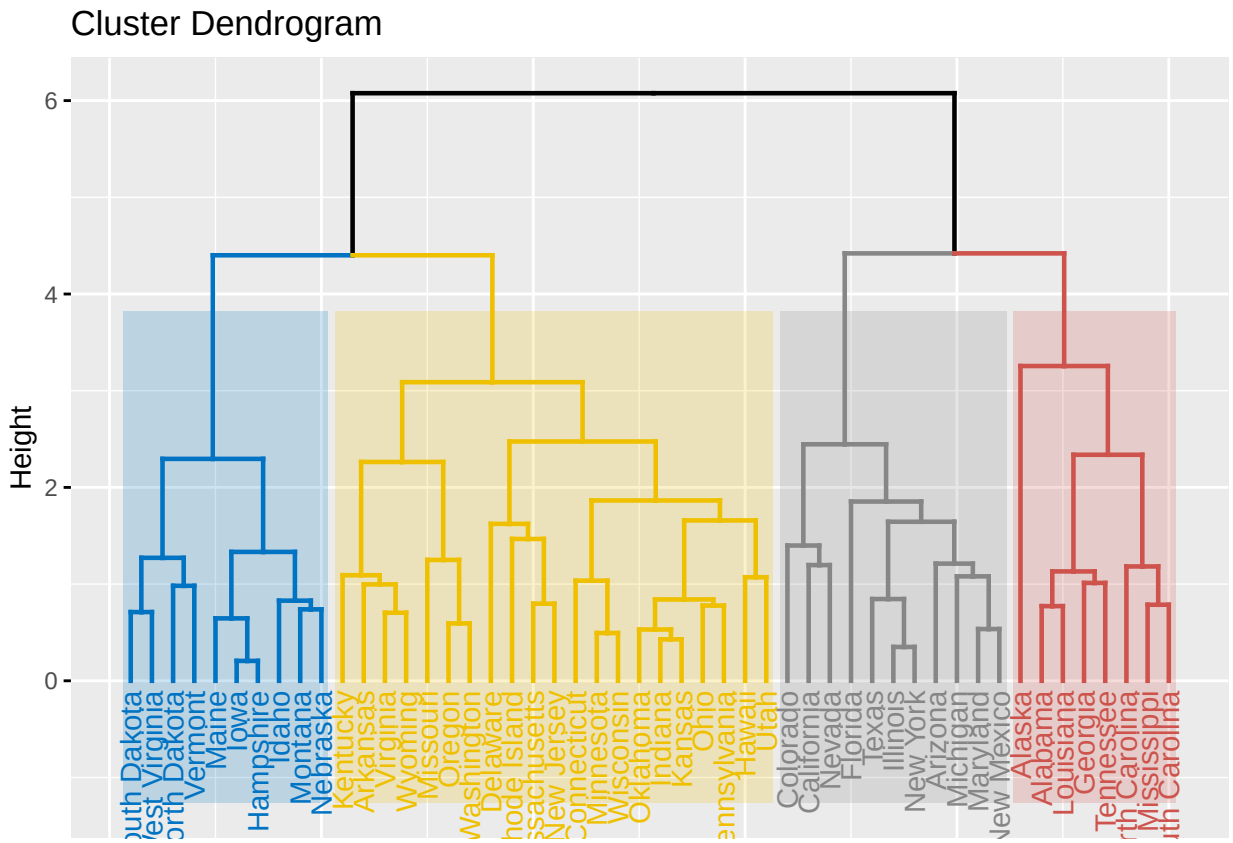
```
fviz_dend(res.hc, cex = 0.8, k=4,
          rect = TRUE,
          k_colors = "jco",
          rect_border = "jco",
          rect_fill = TRUE,
          horiz = TRUE)
```



Aplicar temas Se pueden aplicar diferentes temas del paquete ggplot2 al dendrograma especificando el valor del ggtheme argumento. El valor predeterminado para este argumento es theme_classic(). Los valores permitidos para este argumento incluyen los siguientes temas oficiales de ggplot2 .

ggtheme: theme_gray (), theme_bw (), theme_minimal (), theme_classic (), theme_void (),. . . .

```
fviz_dend(res.hc, cex = 0.8, lwd = 0.8, k = 4,
          rect = TRUE,
          k_colors = "jco",
          rect_border = "jco",
          rect_fill = TRUE,
          ggtheme = theme_gray())
```



Cambiar el tipo de dendrograma El tipo de dendrograma se puede cambiar estableciendo un valor para el type argumento. Los valores permitidos para este argumento son los siguientes:

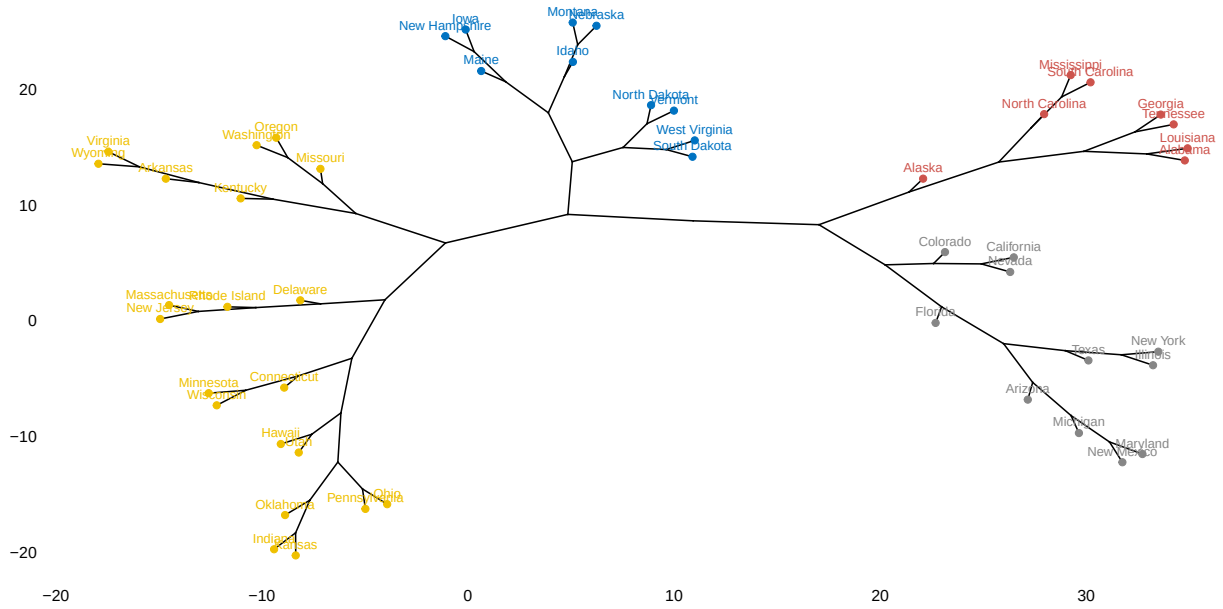
```
#Phylogenic
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
Phylo = fviz_dend(res.hc, cex = 0.8, lwd = 0.8, k = 4,
                  rect = TRUE,
                  k_colors = "jco",
                  rect_border = "jco",
                  rect_fill = TRUE,
                  type = "phylogenic")
Phylo
```



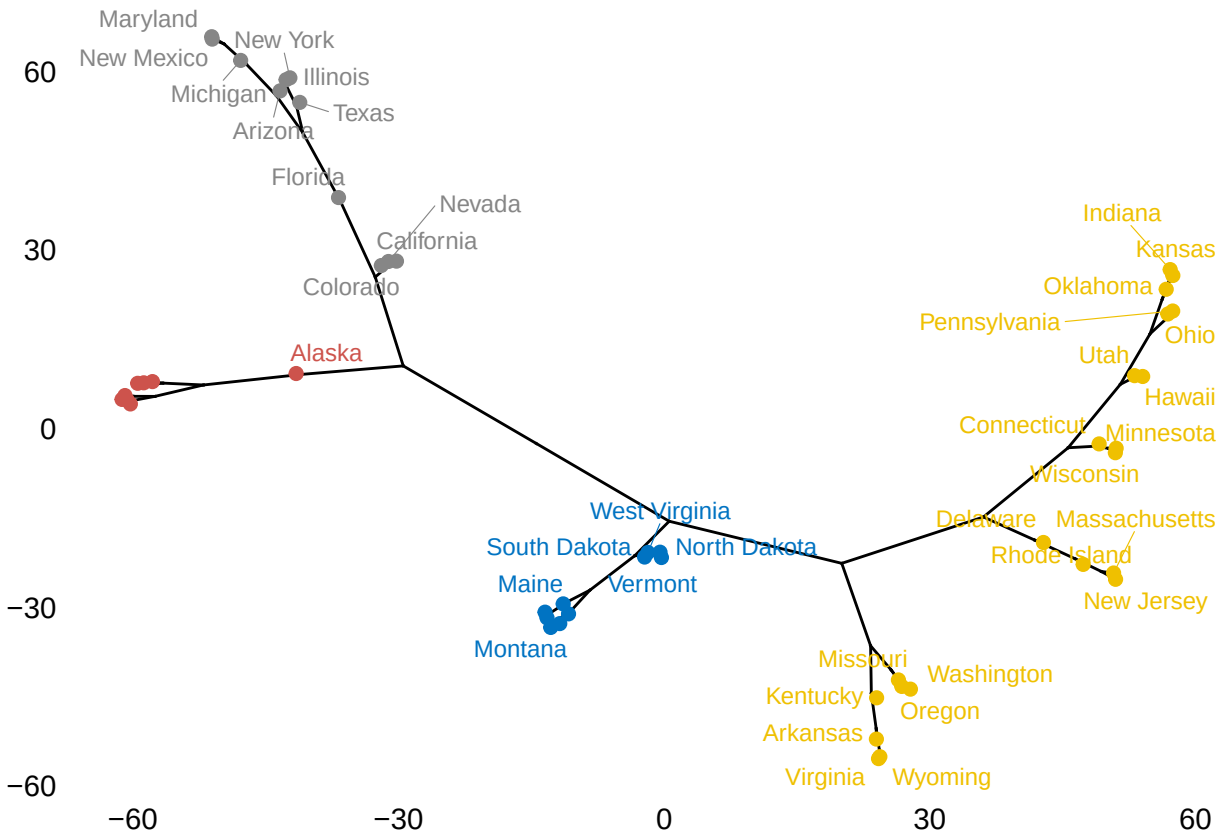

```
# Circular
Circ = fviz_dend(res.hc, cex = 0.8, lwd = 0.8, k = 4,
                 rect = TRUE,
                 k_colors = "jco",
                 rect_border = "jco",
                 rect_fill = TRUE,
                 type = "circular")
```

Diseños filogénicos. Se pueden utilizar diferentes diseños para los árboles filogenéticos. Para hacer esto, establezca un valor para el phylo_layoutargumento. El valor predeterminado para este argumento es layout.auto. Los valores permitidos para este argumento incluyen:

phylo_layout: “Layout.auto”, “layout_with_drl”, “layout_as_tree”, “layout.gem”, “layout.mds” y “layout_with_lgl”

```
fviz_dend(res.hc, cex = 0.8, lwd = 0.8, k = 4,
          rect = TRUE, k_colors = "jco", rect_border = "jco",
          rect_fill = TRUE, type = "phylogenic", repel = TRUE,
# phylo_layout (layout_with_drl)
          phylo_layout = "layout_with_drl")
```

```
## Warning: ggrepel: 11 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
# phylo_layout (layout_as_tree)
          #phylo_layout = "layout_as_tree"
# phylo_layout (layout.gem)
          #phylo_layout = "layout.gem"
# phylo_layout (layout.mds)
          #phylo_layout = "layout.mds"
# phylo_layout (layout_with_lgl)
          #phylo_layout = "layout_with_lgl"
```