Modelos de Soporte No Supervisado

HEATMAPS

Heatmaps

- Un mapa de calor es otra forma de visualizar la agrupación jerárquica.
- También llamada imagen de color falso, donde los valores de los datos se transforman en escala de colores.
- Los mapas de calor nos permiten visualizar simultáneamente grupos de muestras y características.
- Primero la agrupación jerárquica se realiza tanto de las filas como de las columnas de la matriz de datos.
- Las columnas / filas de la matriz de datos se vuelven a ordenar según los resultados del agrupamiento jerárquico, poniendo cerca observaciones similares.
- Los valores altos y los valores bajos son adyacentes en la matriz de datos.
- Finalmente, se aplica un esquema de color para la visualización de la matriz de datos.
- Este tipo de diagrama permite encontrar las variables que definen las características para cada cluster generado.

Heatmaps

Hay un número múltiple de paquetes R y funciones para dibujar mapas de calor estáticos einteractivos, que incluyen:

- heatmap () [R base function, stats package]: dibuja un mapa de calor simple
- heatmap.2 () [gplots R package]: dibuja un mapa de calor mejorado
- pheatmap () [paquete R de pheatmap]: proporciona más control para cambiar la apariencia de heatmaps.
- d3heatmap () [d3heatmap R package]: dibuja un mapa de calor interactivo / seleccionable
- Mapa de calor () [Paquete ComplexHeatmap R / Bioconductor]: dibuja, anota y organiza mapas de calor complejos

Heatmaps: ejemplos

- Usamos los datos de mtcars como un conjunto de datos de demostración.
- Comenzamos estandarizando los datos para hacer variables comparables:

df <- **scale**(mtcars)

Heatmaps: función heatmap()

Se puede usar la función R heatmap () incorporada [en el paquete de estadísticas].

Un formato simplificado es:

heatmap (x, scale = "fila")

x: una matriz numérica

scale: indica si los valores deben estar centrados y escalados, ya sea la dirección de la fila o la dirección de la columna, o ninguno.

Los valores permitidos son:

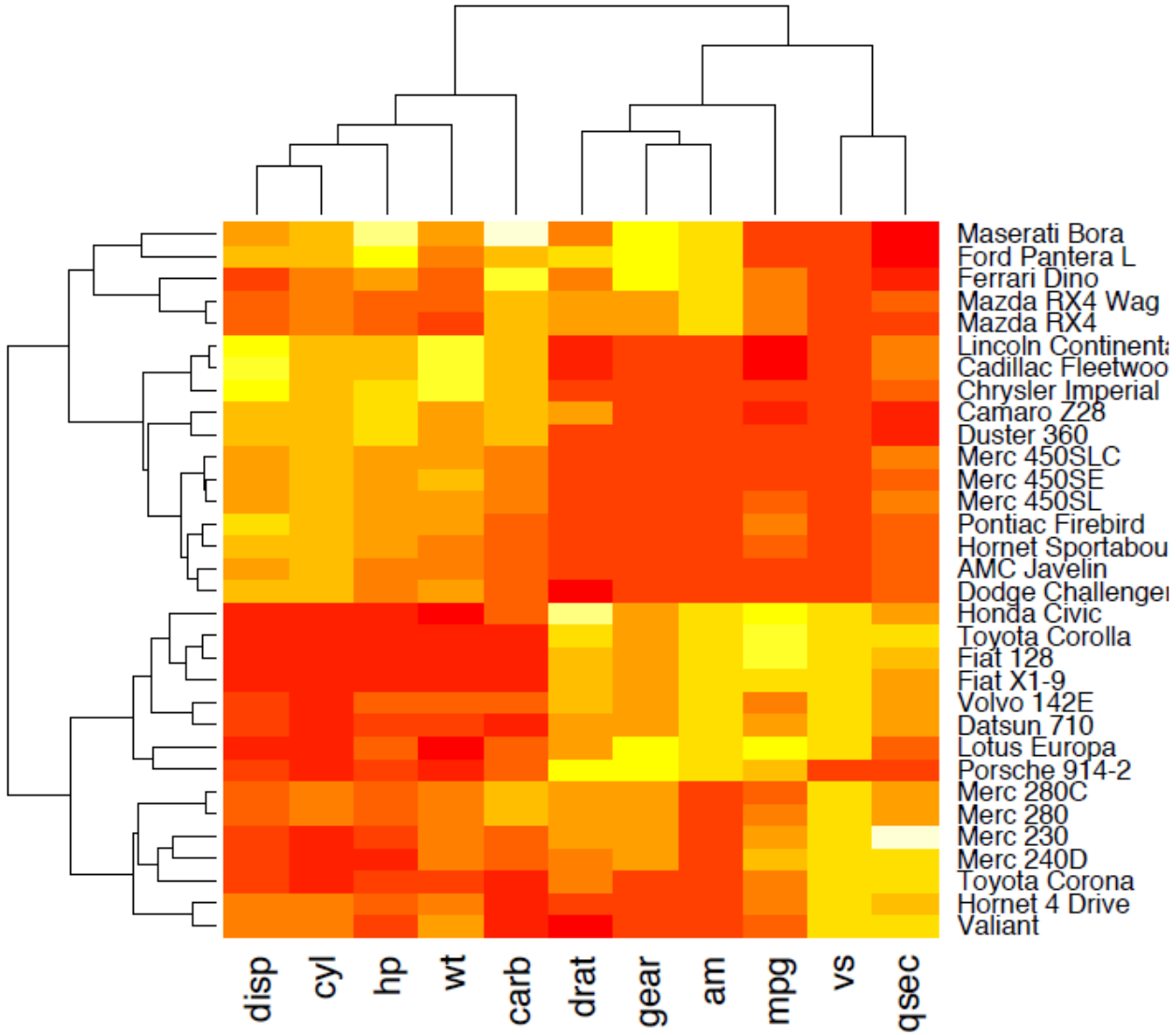
c ("fila", "columna", "ninguno"). El valor predeterminado es "fila".

Heatmaps

 # Gráfica de calor

heatmap(df, scale = "none")

En el gráfico anterior,
los valores altos están en
rojo y los valores bajos en
amarillo.



Heatmaps

Es posible especificar una paleta de colores usando el argumento col, que se puede definir como:

col<- **colorRampPalette**(**c**("red", "white", "blue"))(256)

Utilizando una librería adicional

library("RColorBrewer")
col <- **colorRampPalette**(**brewer.pal**(10, "RdYlBu"))(256)

Heatmaps

- Además, se puede usar el argumento RowSideColors y ColSideColors para diferenciar filas y columnas, respectivamente.

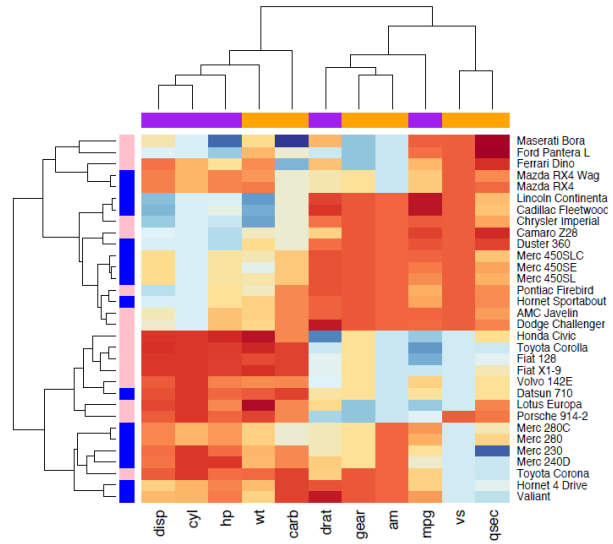
Por ejemplo, en el código R a continuación, se personalizará el mapa de calor de la siguiente manera:

1. Se usa la paleta de colores RColorBrewer para cambiar la apariencia

2. El argumento RowSideColors y ColSideColors se usan para identificar filas y columnas, respectivamente. Los valores esperados para estas opciones son un vector que contiene nombres de colores que especifican las clases para filas / columnas.

Heatmaps

```
#  RColorBrewer
library("RColorBrewer")
col <- colorRampPalette(brewer.pal(10, "RdYlBu"))(256)
heatmap(df, scale = "none", col = col,
RowSideColors = rep(c("blue", "pink"), each = 16),
ColSideColors = c(rep("purple", 5), rep("orange", 6)))
```



Heatmaps

Mapas de calor mejorados: heatmap.2 ()

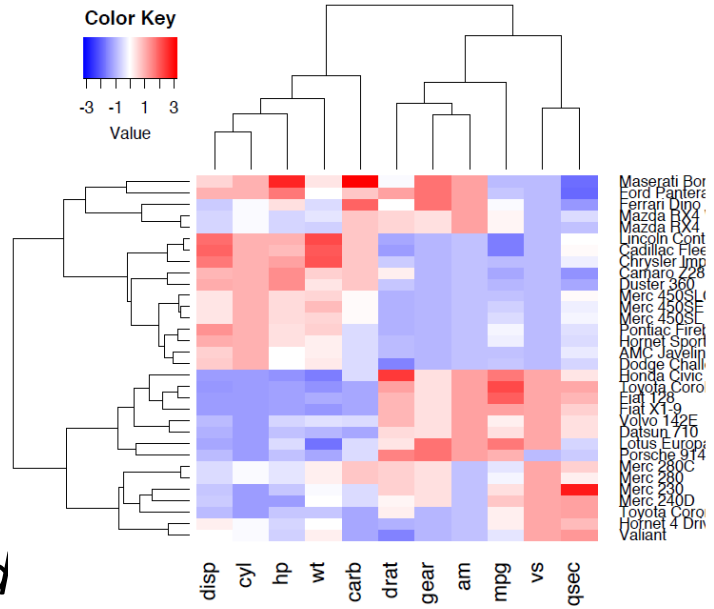
La función heatmap.2 () [en el paquete gplots] proporciona muchas extensiones al estándar de la Función R heatmap () presentada en la sección anterior.

 # install.packages("gplots")

library("gplots")

heatmap.2(df, scale = "none", col = **bluered**(100), trace = "none", density.info = "none")

Heatmaps



Otros argumentos que pueden ser usados:

labRow, labCol

hclustfun: hclustfun = function (x) hclust (x, method = "ward

En el código R anterior, la función bluered () [en el paquete gplots] se usa para generar un conjunto de colores que varía suavemente. También puede usar el siguiente generador de color:

colorpanel (n, bajo, medio, alto)

- n: Número deseado de elementos de color que se generarán

- bajo, medio, alto: colores para usar para los valores más bajo, medio y alto.

redgreen (n), greenred (n), bluered (n) y redblue (n)

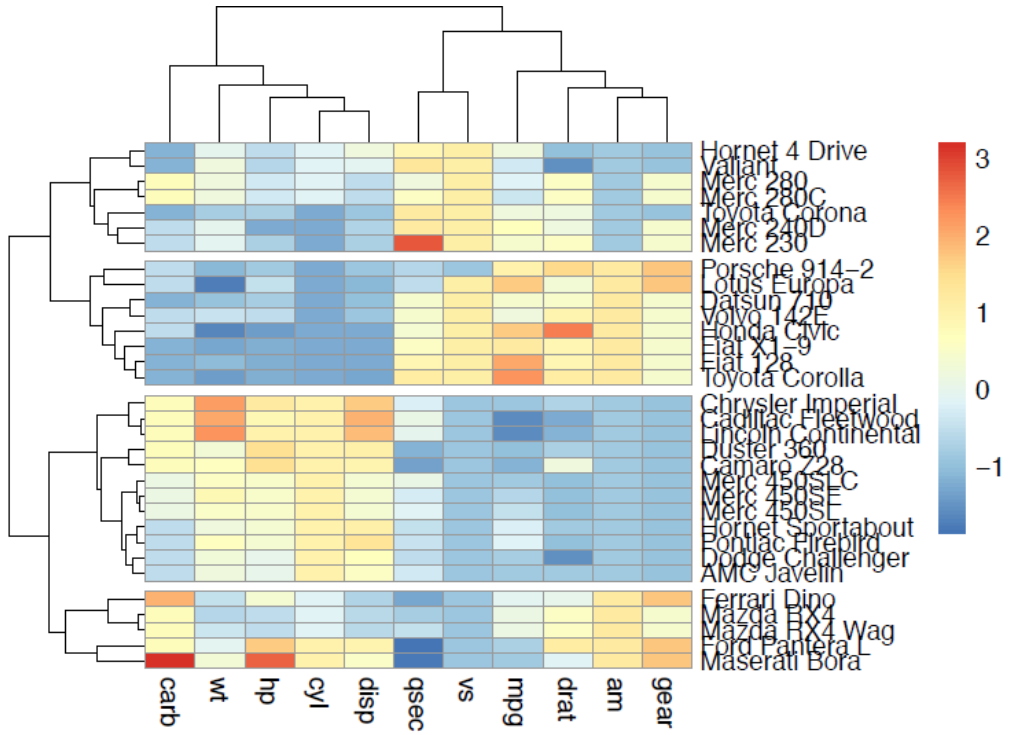
Heatmaps

Mapas de calor mejorados: pheatmap ()
Primero, instale el paquete pheatmap: install.packages ("pheatmap")

library("pheatmap")
pheatmap(df, cutree_rows = 4)

Los argumentos están disponibles para cambiar la métrica de clúster predeterminada ("euclidiana") y Método de linkage ("completo").

- También es posible diferenciar filas y columnas mediante la agrupación variables.



Heatmaps

Mapas interactivos de calor: d3heatmap ()

Primero, instale el paquete d3heatmap: install.packages("d3heatmap")

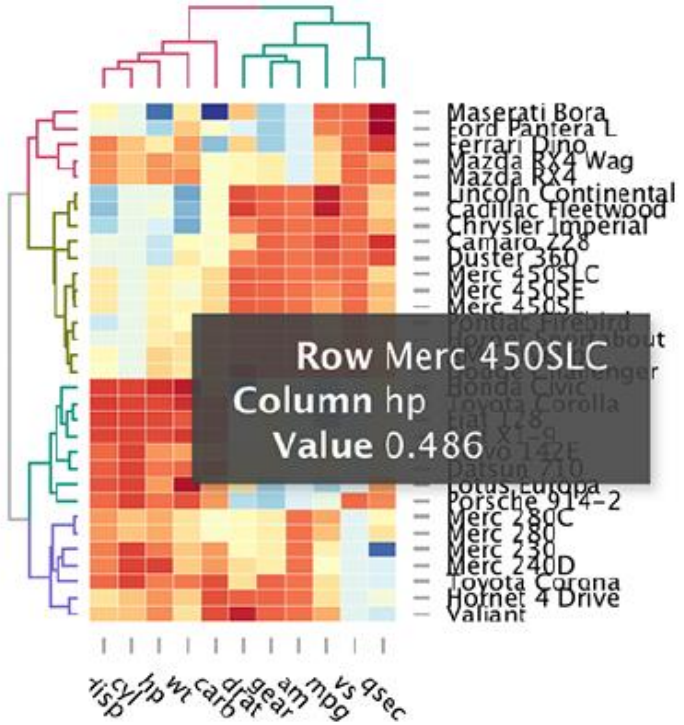
library("d3heatmap")

d3heatmap(**scale**(mtcars), colors = "RdYlBu",

k_row = 4, *# Número de grupos en columnas*

k_col = 2 *# Número de grupos en filas*

)



Heatmaps

Mejora de mapas de calor usando dendextend

- *El paquete dendextend se puede usar para mejorar las funciones de otros paquetes.*
- *Los datos de mtcars se usan en las siguientes secciones.*
- *Comenzaremos por definir el orden y el apariencia para filas y columnas usando dendextend.*
- *Estos resultados se usan en otras funciones de los otros paquetes revisados.*

Heatmaps

El orden y la apariencia de las filas y columnas se pueden definir de la siguiente manera:

```
library(dendextend)
# orden de filas
Rowv <- mtcars %>% scale %>% dist %>% hclust %>% as.dendrogram %>%
set("branches_k_color", k = 3) %>% set("branches_lwd", 1.2) %>%
ladderize

# Order de columnas, debemos trasponer los datos
Colv <- mtcars %>% scale %>% t %>% dist %>% hclust %>% as.dendrogram %>%
set("branches_k_color", k = 2, value = c("orange", "blue")) %>%
set("branches_lwd", 1.2) %>%
ladderize
```

Heatmaps

Utilizamos los argumentos definidos con las otras funciones:

heatmap(scale(mtcars), Rowv = Rowv, Colv = Colv, scale = "none")

library(gplots) heatmap.2(scale(mtcars), scale = "none", col = bluered(100), Rowv = Rowv, Colv = Colv, trace = "none", density.info = "none")

library("d3heatmap") d3heatmap(scale(mtcars), colors = "RdBu", Rowv = Rowv, Colv = Colv)

Heatmaps

ComplexHeatmap es un paquete R / bioconductor, desarrollado por Zuguang Gu, que proporciona una solución flexible para organizar y anotar múltiples heatmaps.

También permite para visualizar la asociación entre diferentes datos de diferentes fuentes.

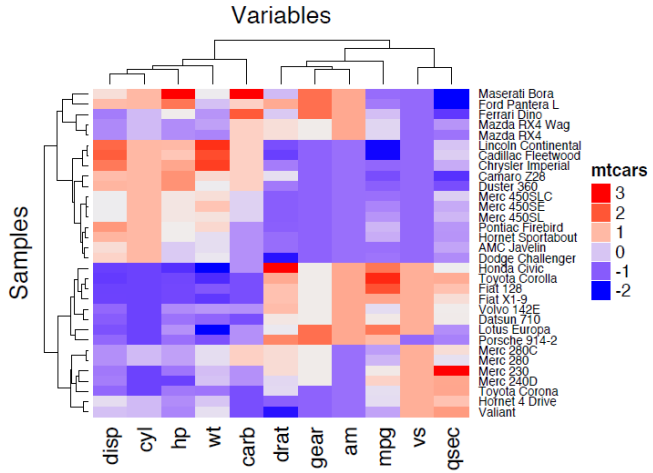
source("https://bioconductor.org/biocLite.R")

biocLite("ComplexHeatmap")

Heatmaps

Mapa de calor simple

Se puede dibujar un mapa de calor simple de la siguiente manera:



library(ComplexHeatmap)

Heatmap(df, name = "mtcars", *# título de la etiqueta*

column_title = "Variables", row_title = "Samples",

row_names_gp = **gpar**(fontsize = 7) *# Tamaño del texto de las filas*

)

Heatmaps

Argumentos adicionales:
1. show_row_names, show_column_names: mostrar nombres de fila y columna nombres, respectivamente. El valor predeterminado es VERDADERO
2. show_row_hclust, show_column_hclust: valor lógico; mostrar la fila y grupos de columnas. El valor predeterminado es VERDADERO
3. clustering_distance_rows, clustering_distance_columns: métrica para clustering: "Euclidiano", "máximo", "manhattan", "canberra", "binario", "minkowski", "Pearson", "spearman", "kendall")
4. clustering_method_rows, clustering_method_columns: métodos de agrupamiento: "Ward.D", "ward.D2", "single", "complete", "average",. . . (ver? hclust).
Para especificar colores personalizados, debe usar la función colorRamp2 () [circlize paquete], de la siguiente manera:

Heatmaps

```
library(circlize)
mycols <- colorRamp2(breaks = c(-2, 0, 2),
                     colors = c("green", "white", "red"))
Heatmap(df, name = "mtcars", col = mycols)
```

Usando la función RColorBrewer

```
library("circlize")
library("RColorBrewer")
Heatmap(df, name = "mtcars",
        col = colorRamp2(c(-2, 0, 2), brewer.pal(n=3, name="RdBu")))
```

Heatmaps

Se puede mejorar la apariencia de los dendogramas

```
library(dendextend)
row_dend = hclust(dist(df)) # row clustering
col_dend = hclust(dist(t(df))) # column clustering
Heatmap(df, name = "mtcars",
        row_names_gp = gpar(fontsize = 6.5),
        cluster_rows = color_branches(row_dend, k = 4),
        cluster_columns = color_branches(col_dend, k = 2))
```

Heatmaps

División de mapa de calor por filas

Se puede dividir el mapa de calor utilizando el algoritmo k-means o una variable de agrupamiento.

- *# Ejemplo dividiendo en dos grupos*

set.seed(2)

Heatmap(df, name = "mtcars", k = 2)

Heatmaps

Para dividir por una variable de agrupación, es necesario definir un vector que contiene esta clasificación

En el siguiente ejemplo usaremos los niveles de la variable de factor cyl [en el conjunto de datos mtcars] para dividir el mapa de calor por filas. Recuerde que la columna cyl corresponde al número de cilindros.

- *# divide por un vector que especifica los grupos (por filas)*

Heatmap(df, name = "mtcars", split = mtcars$cyl,

row_names_gp = **gpar**(fontsize = 7))

- *# dividir combinando multiples variables*

Heatmap(df, name ="mtcars",

split = **data.frame**(cyl = mtcars$cyl, am = mtcars$am))

Heatmaps

Anotaciones en el mapa de calor

La clase de anotación Heatmap se usa para definir la anotación en fila o columna. Un formato simplificado es:

HeatmapAnnotation (df, name, col, show_legend)

df: un data.frame con nombres de columna

nombre: el nombre de la anotación del mapa de calor

col: una lista de colores que contiene mapeo de color a columnas en df

Para el siguiente ejemplo, transpondremos nuestros datos para tener las observaciones en columnas y las variables en filas.

df <- t (df)

Heatmaps

Anotación simple

Un vector, que contiene valores discretos o continuos, se usa para identificar filas o columnas.

Utilizaremos las variables cualitativas cyl (levels = "4", "5" y "8") y am (levels = "0“ y "1"), y la variable continua mpg para identificar columnas.

Para cada una de estas 3 variables, los colores personalizados se definen de la siguiente manera:

Heatmaps

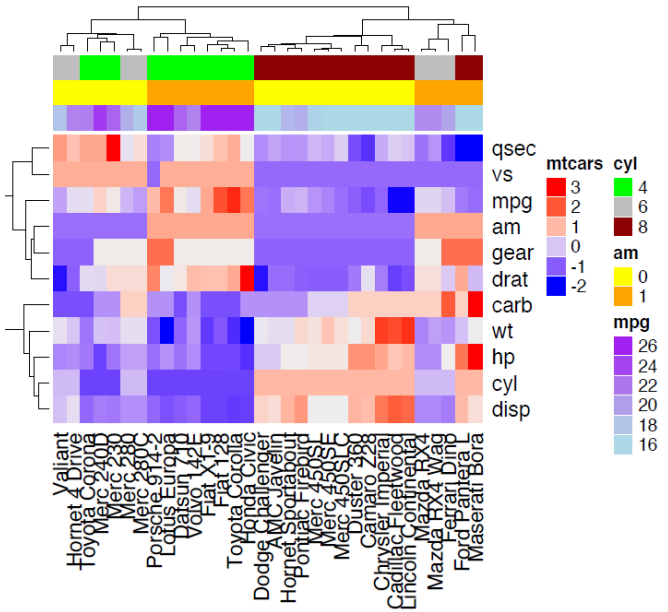
```
# Identificación de valores
annot_df <- data.frame(cyl = mtcars$cyl, am = mtcars$am, mpg = mtcars$mpg)
# Definir los colores por cada nivel para las variables cualitativas
# Definir el gradiente de colores para la variable continua (mpg)
col = list(cyl = c("4" = "green", "6" = "gray", "8" = "darkred"),am = c("0" = "yellow",
"1" = "orange"),
mpg = circlize::colorRamp2(c(17, 25), c("lightblue", "purple")) )
# Crear las anotaciones (indicadores)
ha <- HeatmapAnnotation(annot_df, col = col)
# Combinar las anotaciones en el mapa de calor
Heatmap(df, name = "mtcars", top_annotation = ha)
```

Heatmaps

- *Es posible ocultar la leyenda de la anotación usando el argumento show_legend = FALSE*

ha <- **HeatmapAnnotation**(annot_df, col = col, show_legend = FALSE)

Heatmap(df, name = "mtcars", top_annotation = ha)



Heatmaps

Anotación compleja

En esta sección veremos cómo combinar heatmap y algunos gráficos básicos para mostrar la distribución de datos. Para gráficos de anotación simples, las siguientes funciones se puede usar:

anno_points (), anno_barplot (), anno_boxplot (), anno_density () y anno_histogram ()

Heatmaps

Un ejemplo se muestra a continuación:

Definir algunos gráficos para mostrar la distribución de columnas

.hist = **anno_histogram**(df, gp = **gpar**(fill = "lightblue"))

.density = **anno_density**(df, type = "line", gp = **gpar**(col = "blue"))

ha_mix_top = **HeatmapAnnotation**(hist = .hist, density = .density)

 # Definir algunos gráficos para mostrar la distribución de filas

.violin = **anno_density**(df, type = "violin",

gp = **gpar**(fill = "lightblue"), which = "row")

.boxplot = **anno_boxplot**(df, which = "row")

ha_mix_right = **HeatmapAnnotation**(violin = .violin, bxplt = .boxplot,

which = "row", width = **unit**(4, "cm"))

 # Combina la anotación con el mapa de calor

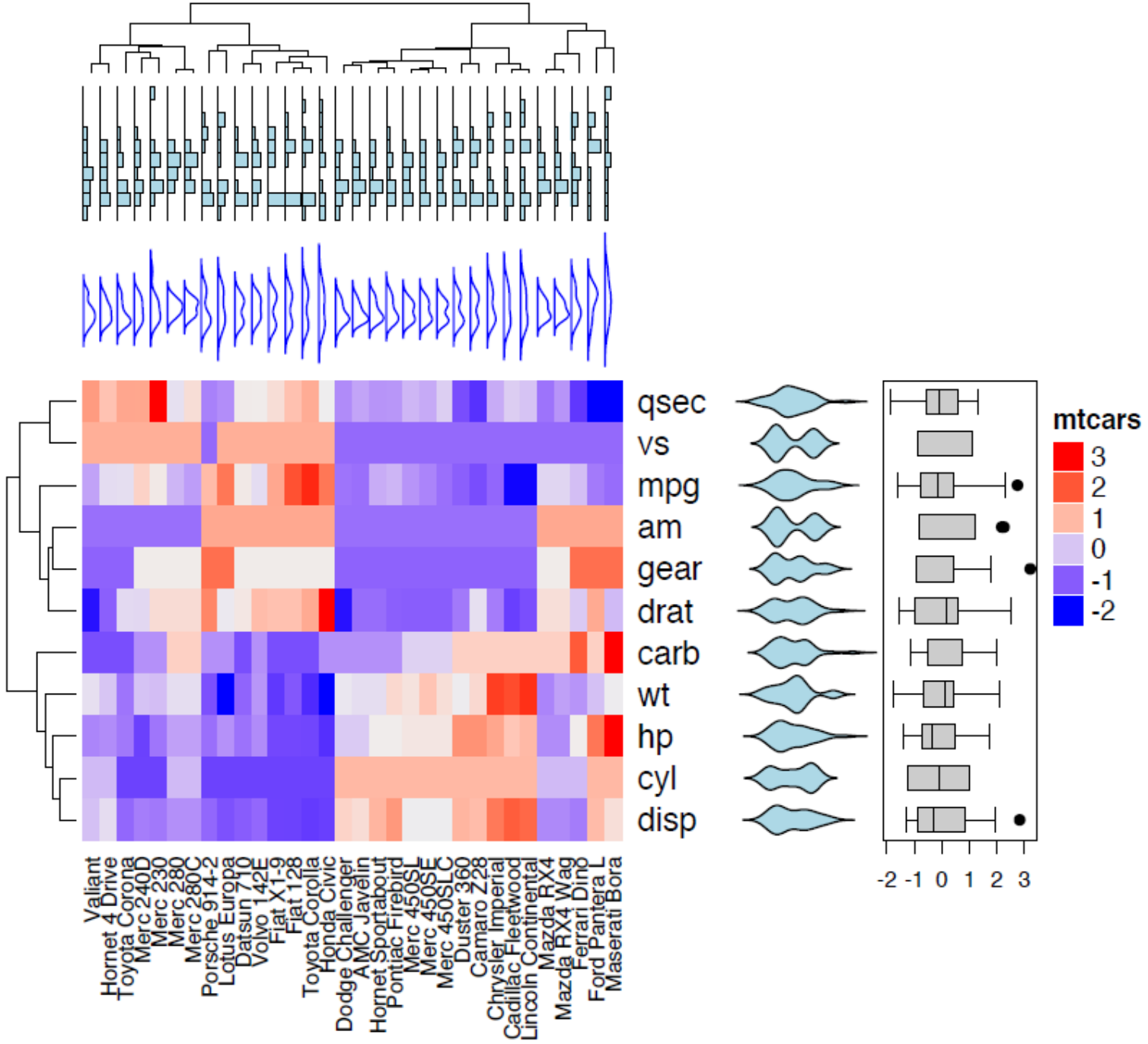
Heatmap(df, name = "mtcars",

column_names_gp = **gpar**(fontsize = 8),

top_annotation = ha_mix_top,

top_annotation_height = **unit**(3.8, "cm")) + ha_mix_right

Heatmaps


qsec
vs
mpg
am
gear
drat
carb
wt
hp
cyl
disp
mtcars
3
2
1
0
-1
-2
-2 -1 0 1 2 3
Valiant
Hornet 4 Drive
Toyota Corona
Merc 240D
Merc 230
Merc 280
Merc 280C
Porsche 914-2
Lotus Europa
Datsun 710
Volvo 142E
Fiat X1-9
Fiat 128
Toyota Corolla
Honda Civic
Dodge Challenger
AMC Javelin
Hornet Sportabout
Pontiac Firebird
Merc 450SL
Merc 450SE
Merc 450SLC
Duster 360
Camaro Z28
Chrysler Imperial
Cadillac Fleetwood
Lincoln Continental
Mazda RX4
Mazda RX4 Wag
Ferrari Dino
Ford Pantera L
Maserati Bora

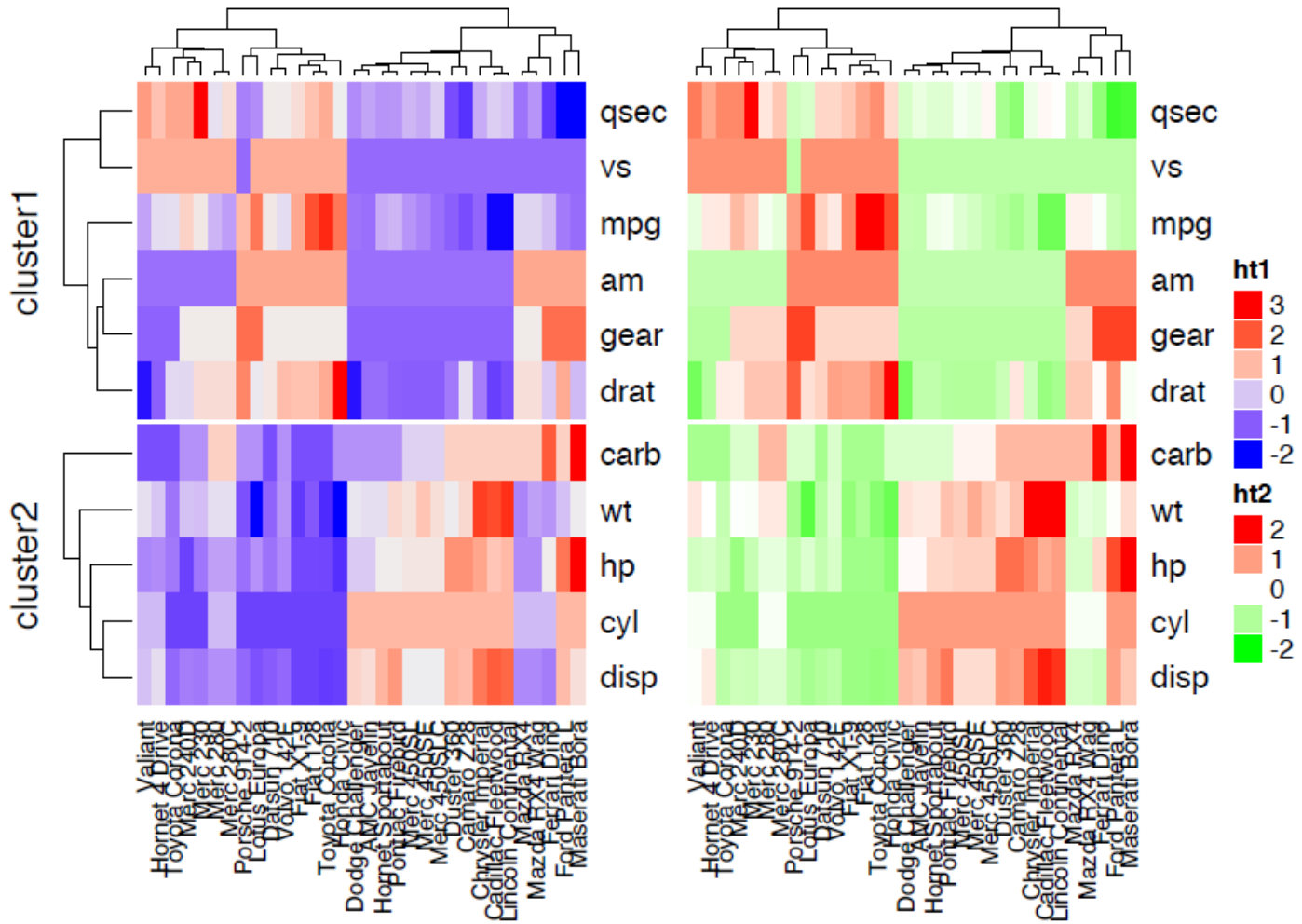
Heatmaps

Múltiples heatmaps se pueden organizar de la siguiente manera:

```
# Heatmap 1
ht1 = Heatmap(df, name = "ht1", km = 2,
column_names_gp = gpar(fontsize = 9))
# Heatmap 2
ht2 = Heatmap(df, name = "ht2",
col = circlize::colorRamp2(c(-2, 0, 2), c("green", "white", "red")),
column_names_gp = gpar(fontsize = 9))
# Combinar los dos heatmaps
```

- ht1 + ht2

Heatmaps


cluster1
cluster2
qsec
vs
mpg
am
gear
drat
carb
wt
hp
cyl
disp
ht1
3
2
1
0
-1
-2
ht2
2
1
0
-1
-2
Valiant
Hornet 4 Drive
Toyota Corona
Merc 240D
Merc 230
Merc 280
Merc 280C
Porsche 914-2
Lotus Europa
Datsun 710
Volvo 142E
Fiat X1-9
Fiat 128
Toyota Corolla
Honda Civic
Dodge Challenger
AMC Javelin
Hornet Sportabout
Pontiac Firebird
Merc 450SE
Merc 450SL
Merc 450SLC
Duster 360
Camaro Z28
Chrysler Imperial
Cadillac Fleetwood
Lincoln Continental
Mazda RX4
Mazda RX4 Wag
Ferrari Dino
Ford Pantera L
Maserati Bora

Heatmaps

La función draw () se puede usar para personalizar la apariencia de la imagen final:

```
draw(ht1 + ht2,
        row_title = "Two heatmaps, row title",
        row_title_gp = gpar(col = "red"),
        column_title = "Two heatmaps, column title",
        column_title_side = "bottom",
# Brecha entre mapas de calor
        gap = unit(0.5, "cm"))
```

Heatmaps

Aplicación a la matriz de expresión génica

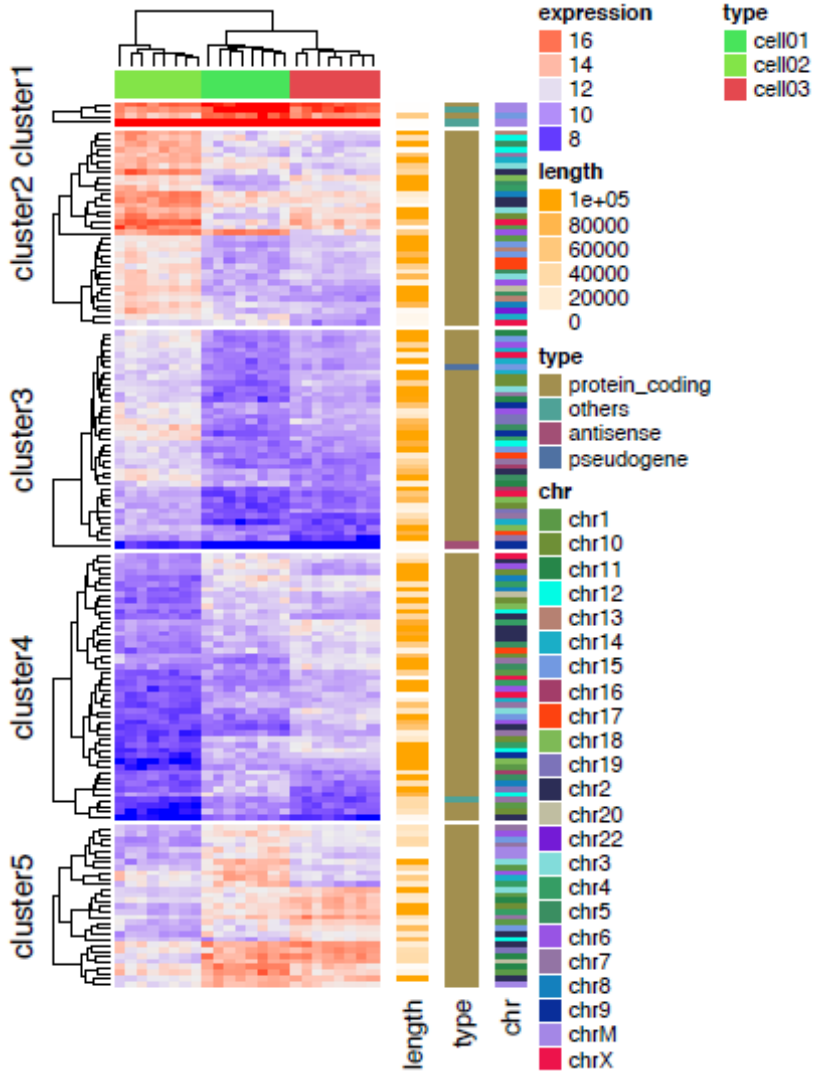
En los datos de expresión génica, las filas son genes y las columnas son muestras. Más información acerca de los genes se pueden unir después de la expresión mapa de calor, como la longitud del gen y tipo de genes

```
expr <- readRDS(paste0(system.file(package = "ComplexHeatmap"),
        "/extdata/gene_expression.rds"))
mat <- as.matrix(expr[, grep("cell", colnames(expr))])
        type <- gsub("s\\d+_", "", colnames(mat))
ha = HeatmapAnnotation(df = data.frame(type = type))
```

Heatmaps

```
Heatmap(mat, name = "expression", km = 5, top_annotation = ha,
        top_annotation_height = unit(4, "mm"),
        show_row_names = FALSE, show_column_names = FALSE) +
Heatmap(expr$length, name = "length", width = unit(5, "mm"),
        col = circlize::colorRamp2(c(0, 100000), c("white", "orange"))) +
Heatmap(expr$type, name = "type", width = unit(5, "mm")) +
Heatmap(expr$chr, name = "chr", width = unit(5, "mm"),
        col = circlize::rand_color(length(unique(expr$chr))))
```

Heatmaps


cluster1
cluster2
cluster3
cluster4
cluster5
length
type
chr
expression
16
14
12
10
8
length
1e+05
80000
60000
40000
20000
0
type
protein_coding
others
antisense
pseudogene
chr
chr1
chr10
chr11
chr12
chr13
chr14
chr15
chr16
chr17
chr18
chr19
chr2
chr20
chr22
chr3
chr4
chr5
chr6
chr7
chr8
chr9
chrM
chrX
type
cell01
cell02
cell03

Heatmaps

- **Describimos muchas funciones para dibujar mapas de calor en R (de básico a complejo mapas de calor).**

- **Se puede producir un mapa de calor básico utilizando la función base R heatmap () o la función heatmap.2 () [en el paquete gplots].**

- **La función pheatmap (), en el paquete del mismo nombre, crea mapas de calor bonitos, donde los que tienen un mejor control sobre algunos parámetros gráficos, como el tamaño de la celda.**

- **La función Heatmap () [en el paquete ComplexHeatmap] nos permite dibujar, dibujar, anotar y organizar mapas de calor complejos.**