Heatmaps en R (con datos no estandarizados)

Laboratorio 37

Nombre de la alumna: Laura Araceli Guerrero Herrera.

Programa: Doctorado en Ciencias Económico Administrativas.

Semestre: Tercero.

Catedrática: Dra. Carla Carolina Pérez Hernández.

Asignatura: Temas Selectos I: Estadística para las Ciencias Económico Administrativas (Complejidad Económica).

Fecha: 9 de marzo de 2022.

LAGH_LAB37

Araceli Guerrero

9/3/2022

Práctica de Laboratorio 37 - Heatmaps en R (con datos no estandarizados)

Algoritmo modificado por Araceli Guerrero Herrera (ZuRy) :3

Paso 1. Buscar información sobre la base de datos de características de autos disponible en R.

```
?mtcars
```

```
## starting httpd help server ... done
```

Paso 2. Visualizar la los primeros datos de la base.

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Paso 3. Identificar el tipo de datos.

```
class(mtcars)
```

```
## [1] "data.frame"
```

Para generar un Heatmap se requiere una matriz de datos de origen.

Paso 4. Transformar el dataframe en una matriz.

```
mtcars_matrix <- data.matrix(mtcars)
```

Paso 5. Visualizar los primeros datos de la matriz generada.

```
head(mtcars_matrix)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

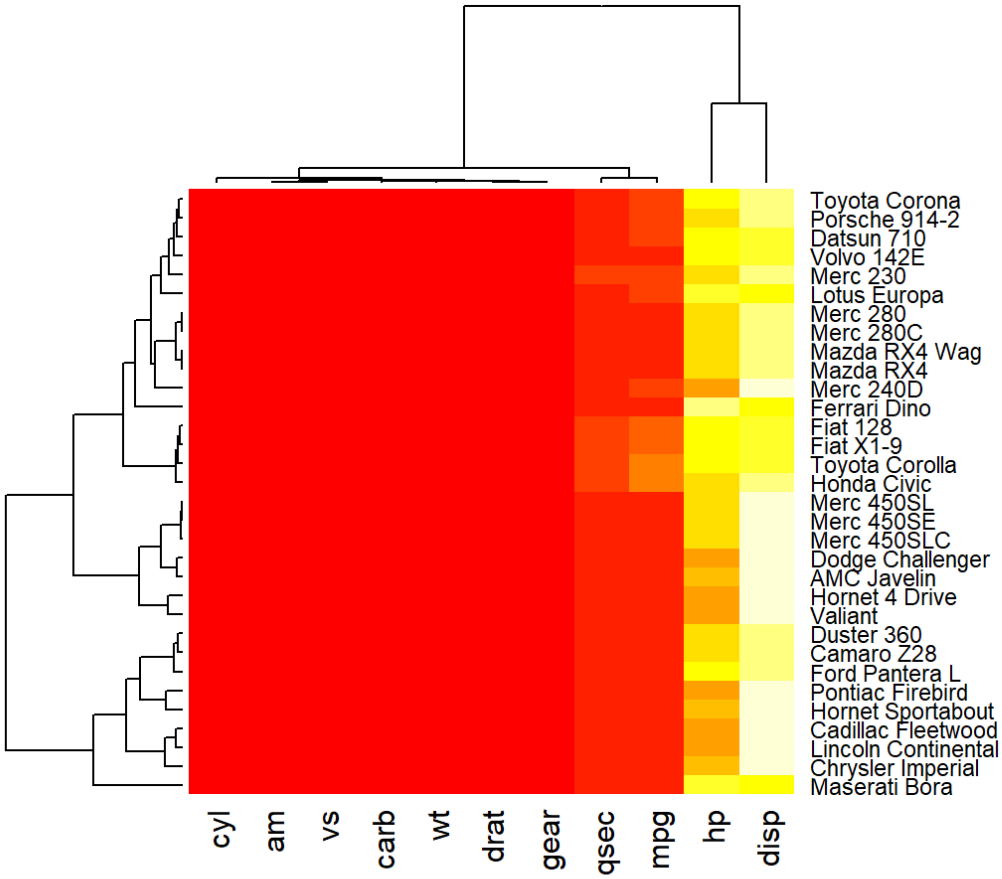
Paso 6. Identificar el tipo de datos.

```
class(mtcars_matrix)
```

```
## [1] "matrix"
```

Paso 7. Generar heatmap de la matriz.

```
heatmap(mtcars_matrix)
```



¿Se parece a lo que esperabas? En el heatmap generado, el escalado de los colores se da con base en los renglones.

Se requiere el escalado de las columnas (que contienen las características de los autos).

Paso 8. Observar la página de ayuda de la función y lea la descripción del scale argument en particular.

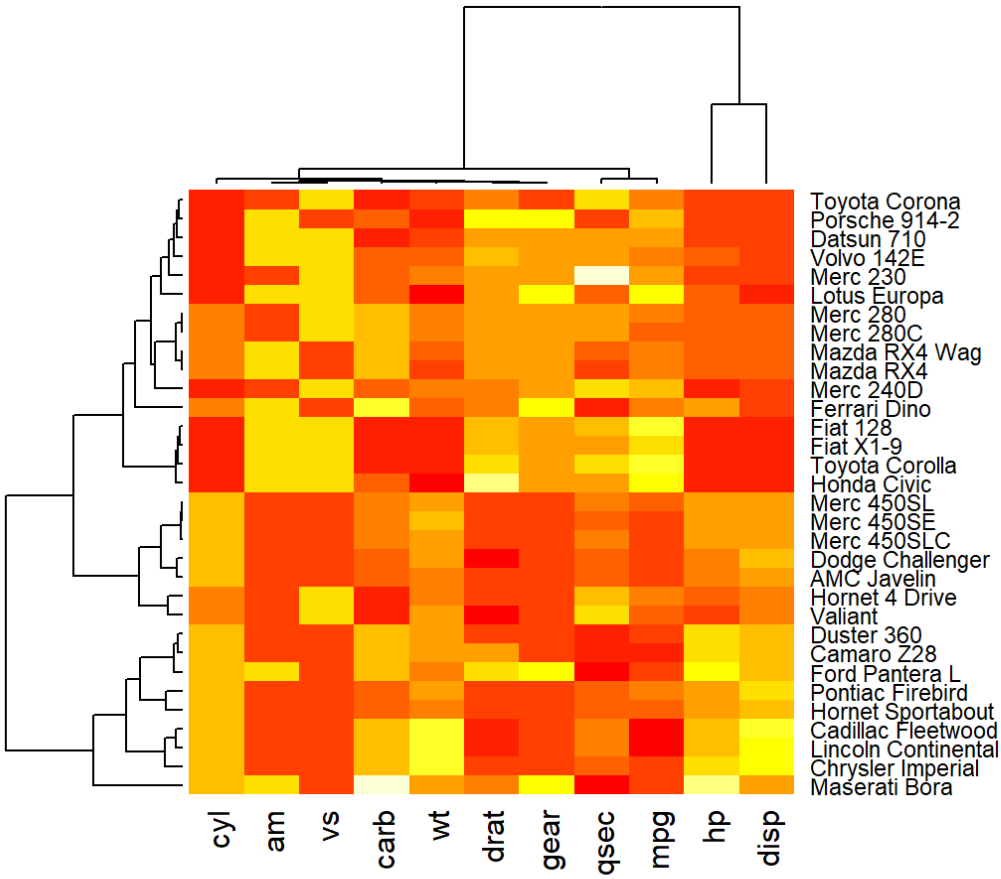
```
?heatmap
```

La escala es importante: los valores deben centrarse y escalarse en filas o columnas.

En nuestro caso, queremos visualizar altibajos en cada variable, que están en columnas.

Paso 9. Generar heatmap con el escalado en las columnas.

```
heatmap(mtcars_matrix, scale = "column")
```



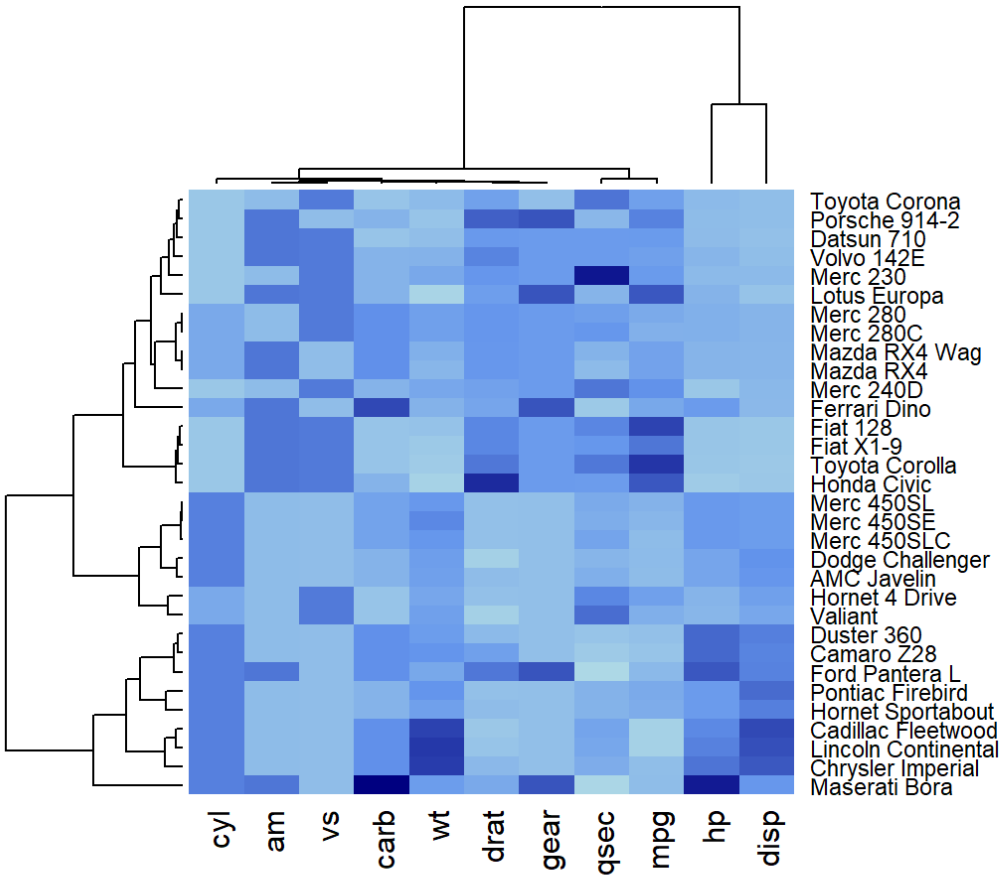
El heatmap generado con escalamiento por columnas, si bien se parece más a lo esperado, la paleta de colores resulta contraintuitivo, pues lo más rojo indica valores mínimos, mientras que lo blanco los máximos.

Paso 10. Hacer una paleta de colores propia.

```
colores_blue <- colorRampPalette(c("lightblue", "cornflowerblue", "navyblue"))(256
)
```

Paso 11. Generar heatmap con la nueva paleta de colores creada.

```
heatmap(mtcars_matrix,
        scale = "column",
        col = colores_blue)
```

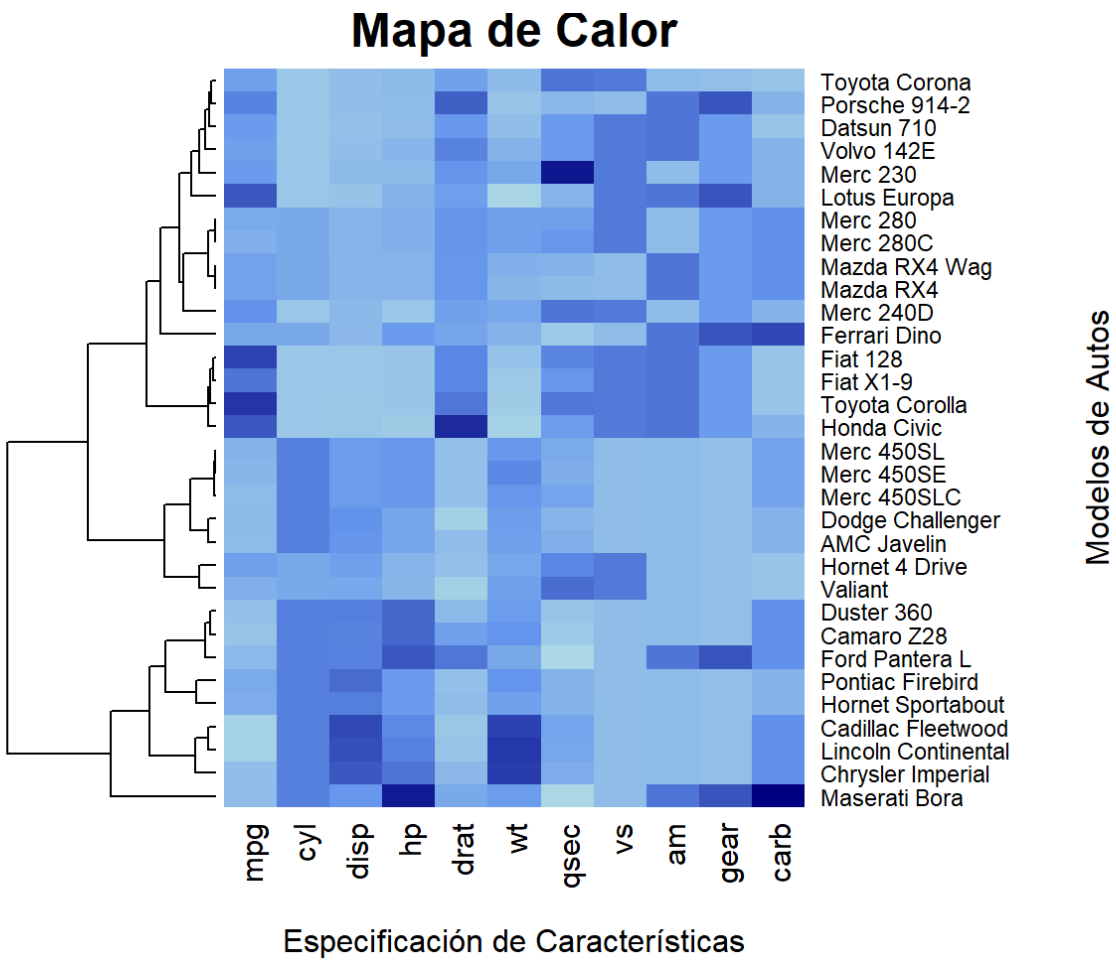


Con AMBOS dendogramas por default, se ordenan las variables por cluster de pertenencia.

El dendrograma de columna realmente no tiene sentido para este conjunto de datos.

Paso 12. Eliminar dendograma de las columnas con Colv=NA.

```
heatmap(mtcars_matrix,
        scale = "column",
        col = colores_blue,
        Colv = NA,
        margins = c(5,10),
        xlab = "Especificación de Características",
        ylab = "Modelos de Autos",
        main = "Mapa de Calor")
```



Paso 13. Visualizar los encabezados de las columnas.

```
colnames(mtcars_matrix)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

Se respeta el orden de las columnas, porque se borró el dendrograma de las columnas, ya no se ordena por clúster.

En cambio, no se respeta el orden de los renglones, porque estos sí tienen clusteo.

Paso 14. Para cambiar los colores del heatmap, se debe intalar y ejecutar la paquetería de viridis.

install.packages(“viridis”)

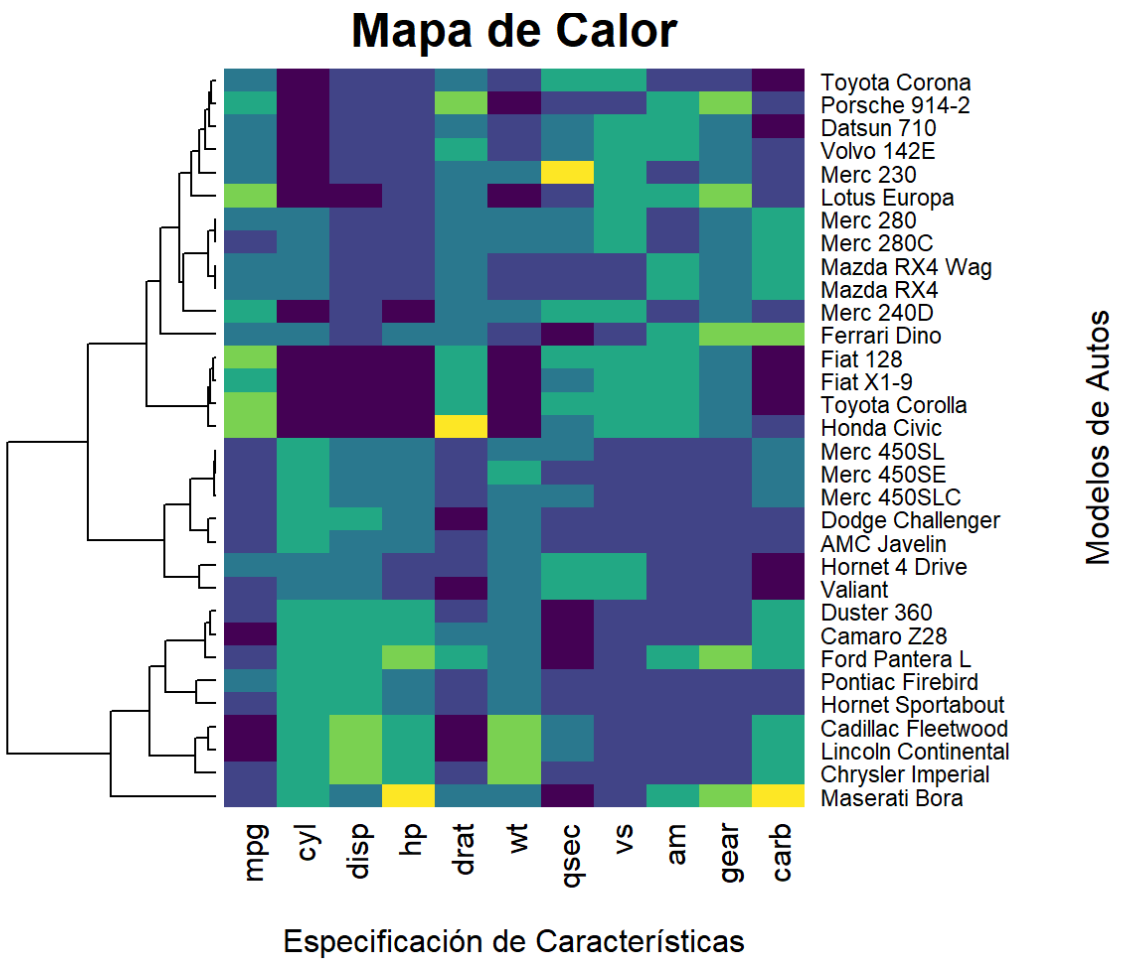
```
library(viridis)
```

```
## Loading required package: viridisLite
```

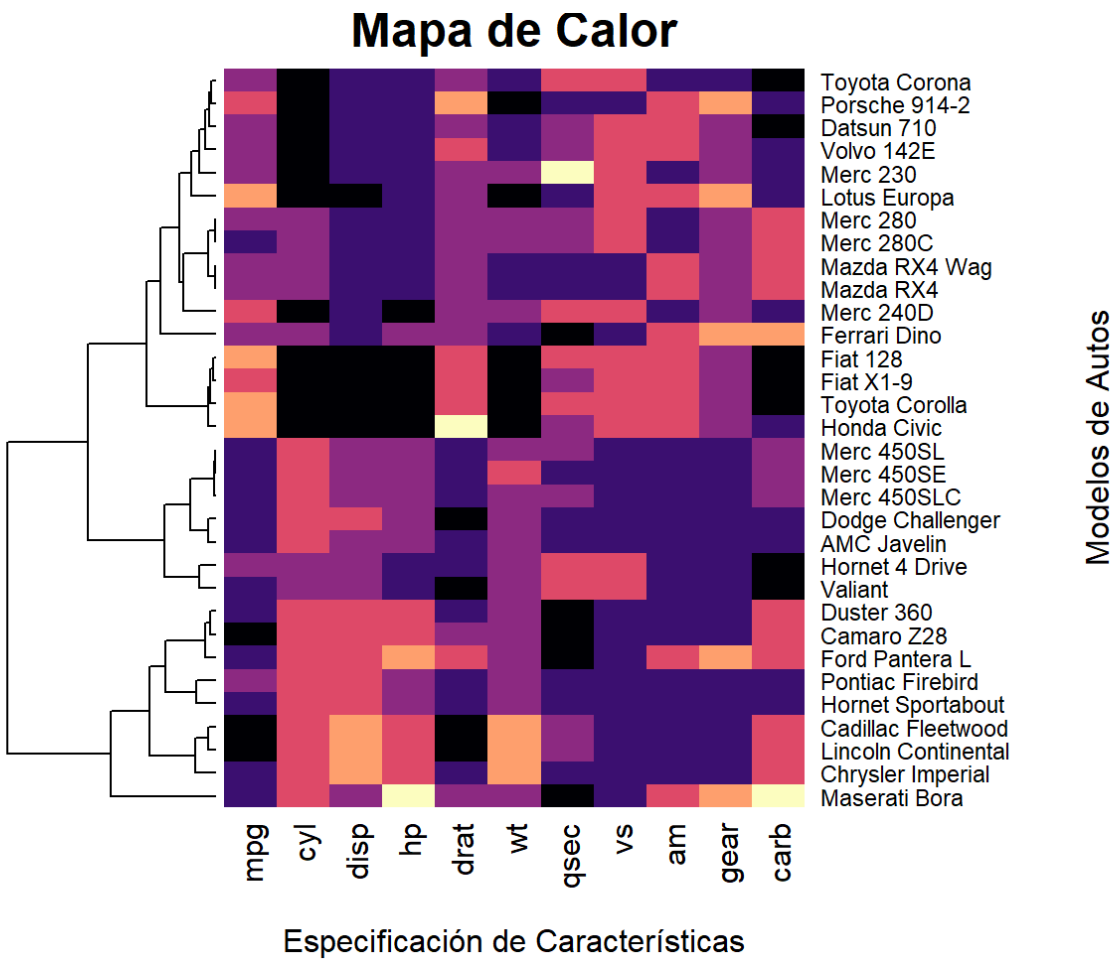
Paso 15. Cambiar el color del heatmap con paletas viridis, magma, plasma, cividis, inferno.

```
heatmap(mtcars_matrix,
        scale = "column",
        col = viridis_pal(option = "viridis") (6),
        Colv = NA,
        margins = c(5,10),
        xlab = "Especificación de Características",
```

```
        ylab = "Modelos de Autos",
        main = "Mapa de Calor")
```

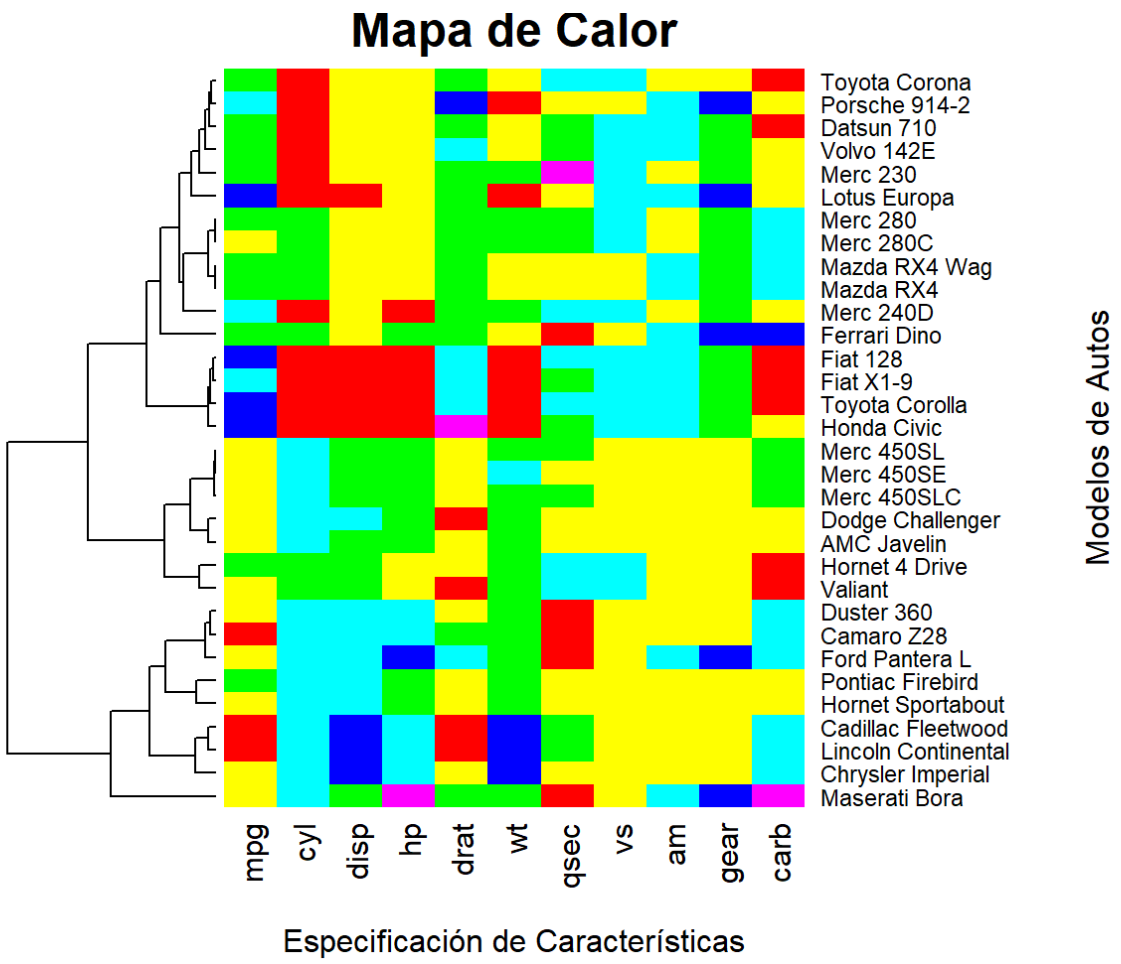



```
heatmap(mtcars_matrix,
        scale = "column",
        col = viridis_pal(option = "magma") (6),
        Colv = NA,
        margins = c(5,10),
        xlab = "Especificación de Características",
        ylab = "Modelos de Autos",
        main = "Mapa de Calor")
```



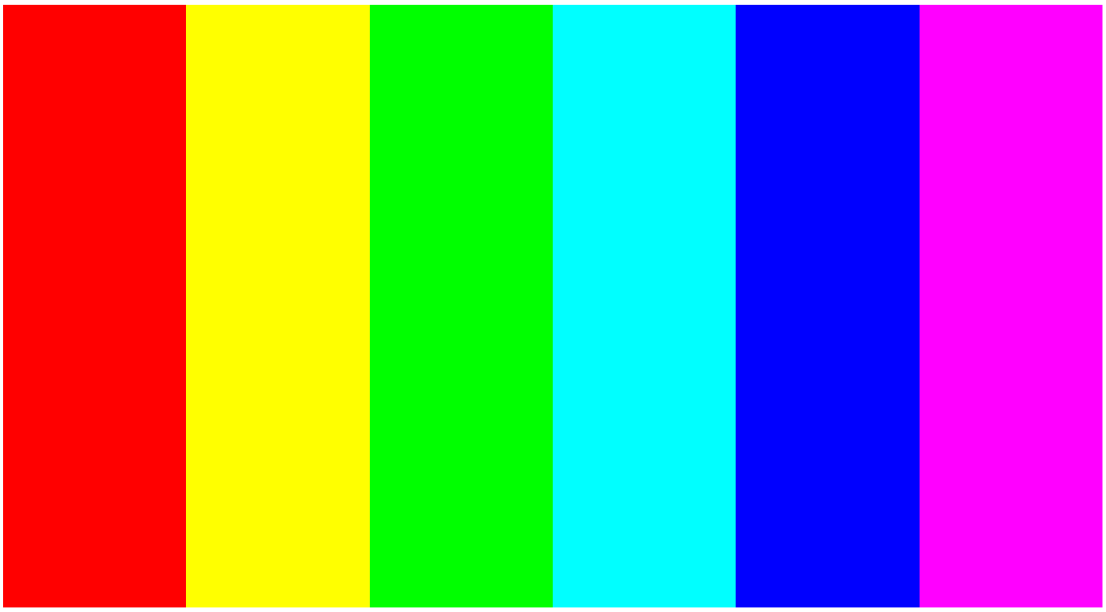
Paso 16. Cambiar color del heatmap usando las paletas por defecto: rainbow, heat.colors, terrain.colors, topo.colors, cm.colors.

```
heatmap(mtcars_matrix,
        scale = "column",
        col = rainbow (6),
        Colv = NA,
        margins = c(5,10),
        xlab = "Especificación de Características",
        ylab = "Modelos de Autos",
        main = "Mapa de Calor")
```



Paso 17. Identificar los valores altos y bajo en el heatmap rainbow.

```
image(1:6,1,as.matrix(1:6), col = rainbow (6), xlab="Leyenda", ylab="", xaxt="n",
 yaxt="n", bty="n")
```



Leyenda

Estandarización de datos.

Paso 1. Generar un objeto llamado datos a partir de la matriz original.

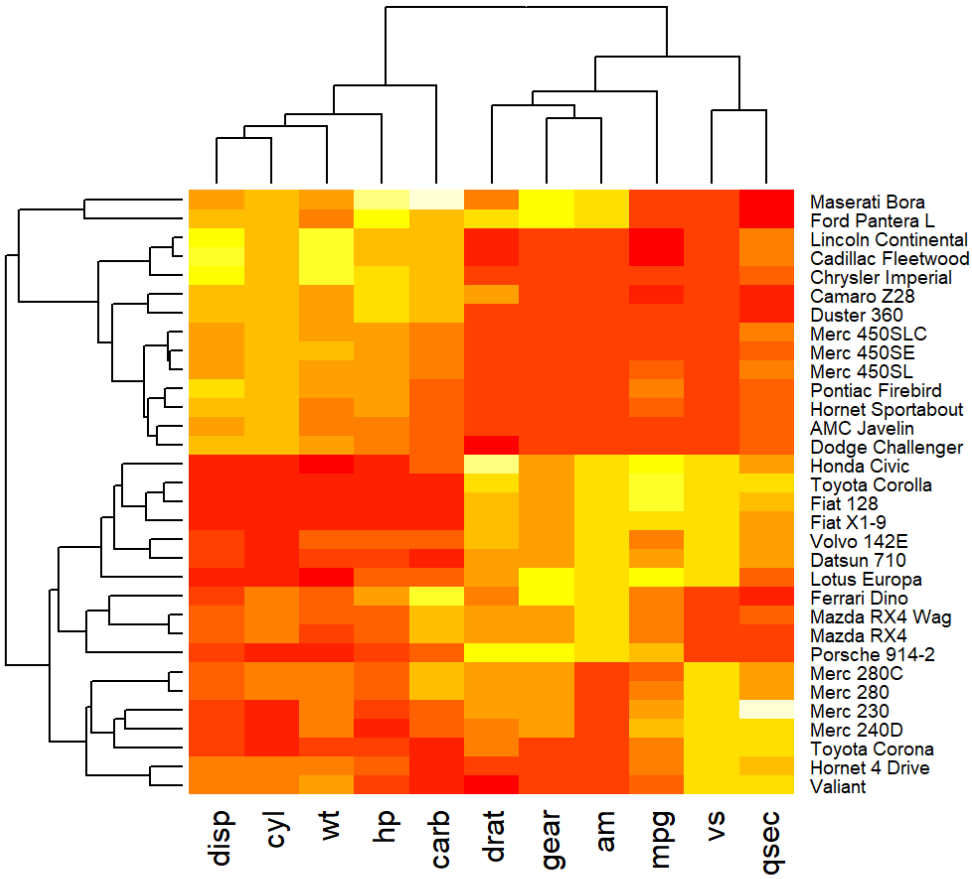
```
datos <- mtcars
```

Paso 2. Para que las variables sean comparables bajo un mismo esquema de colores se estandarizan.

```
datos <- scale(datos)
```

Paso 3. Generar heatmap con los datos escalados.

```
heatmap(x = datos, scale = "none",
        distfun = function(x){dist(x, method = "euclidean")},
        hclustfun = function(x){hclust(x, method = "average")},
        cexRow = 0.7)
```

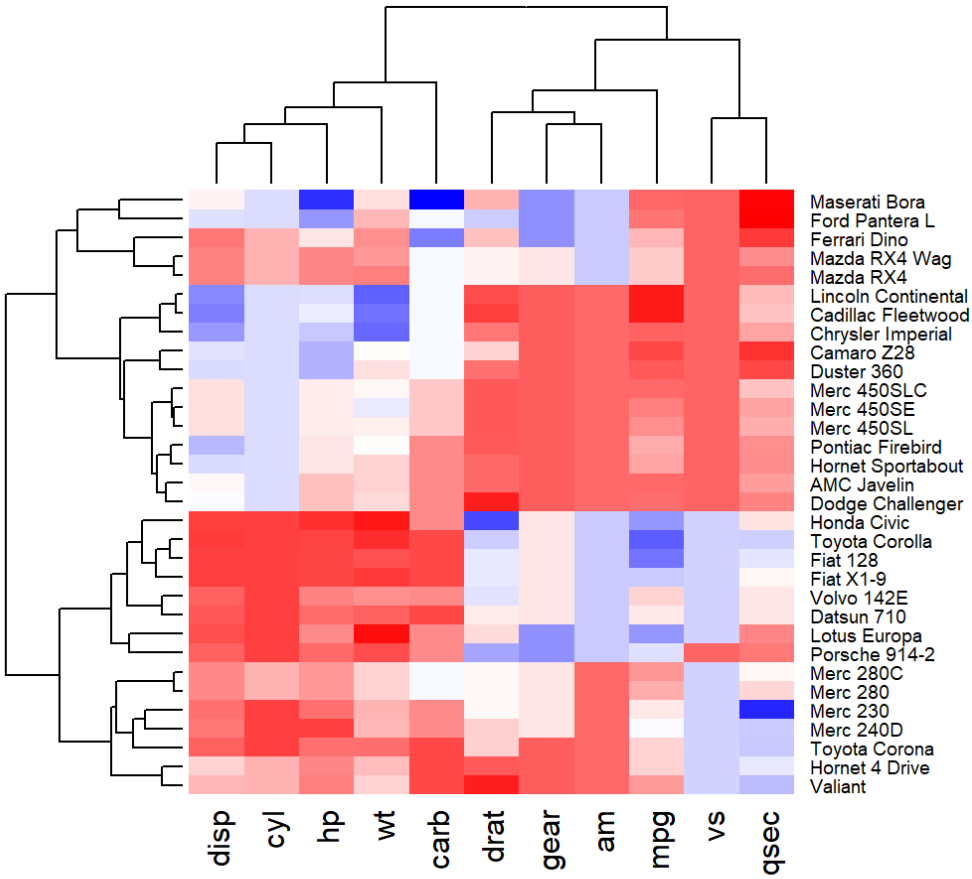


Paso 4. Crear una paleta de colores.

```
colores1 <- colorRampPalette(c("red", "white", "blue"))(256)
```

Paso 5. Generar heatmap con la nueva paleta de colores.

```
heatmap(x = datos, scale = "none", col = colores1, cexRow = 0.7)
```



Paso 6. Usar la paleta de color viridis.

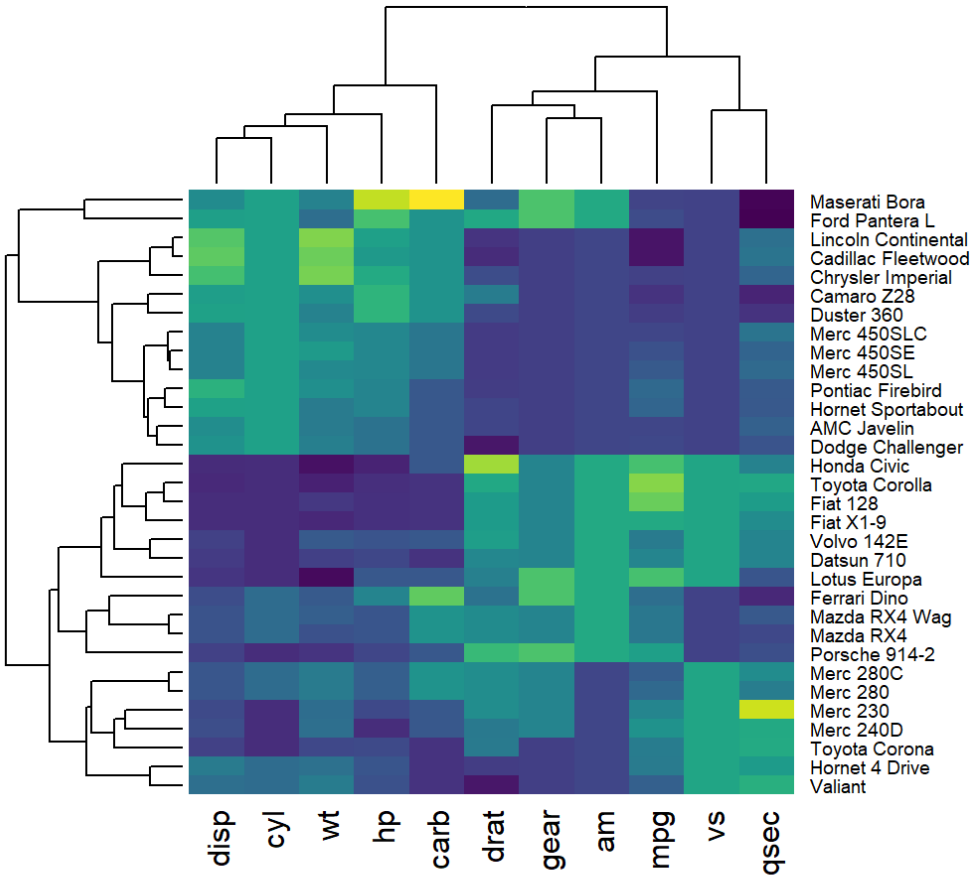
```
library(viridis)
```

Paso 7. Crear una nueva paleta de colores.

```
colores2 <- viridis(256)
```

Paso 8. Generar heatmap con la otra paleta de colores.

```
heatmap(x = datos, scale = "none", col = colores2,
        distfun = function(x){dist(x, method = "euclidean")},
        hclustfun = function(x){hclust(x, method = "average")},
        cexRow = 0.7)
```

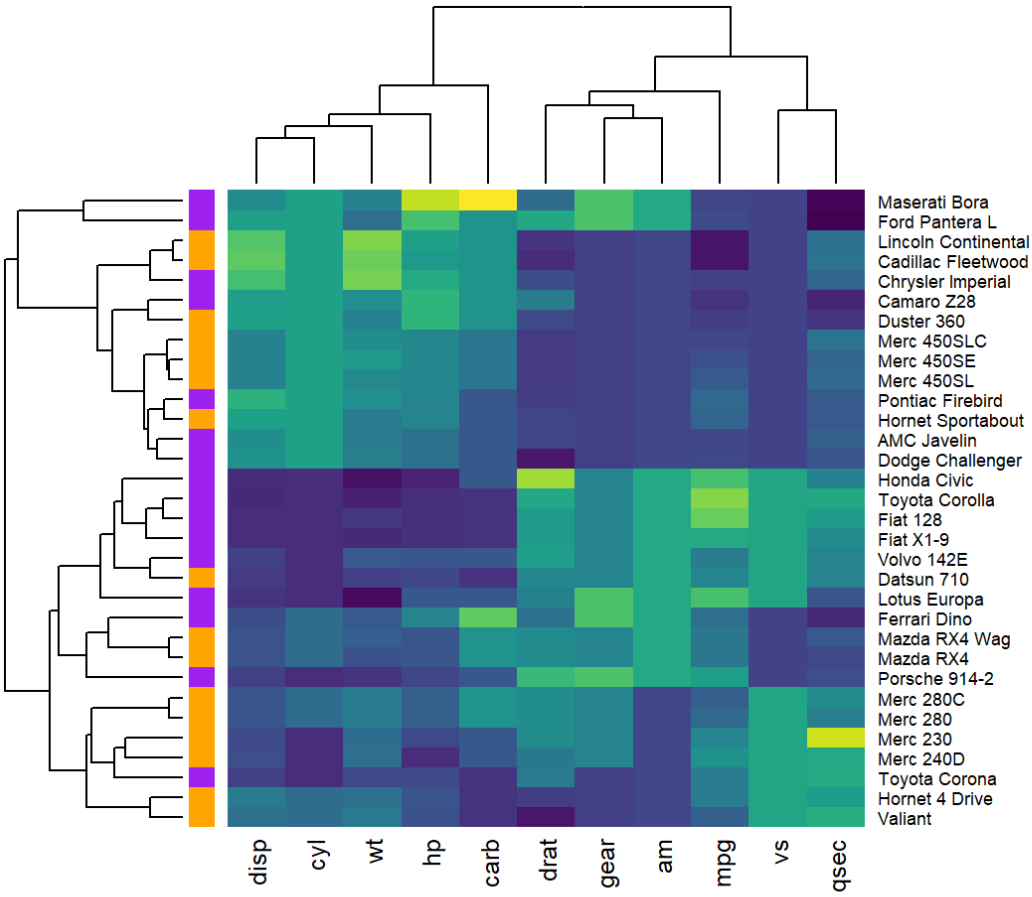


Es posible añadir información adicional (annotate) en las filas o columnas con los argumentos RowSideColors y ColSideColors.

Por ejemplo, supóngase que los primeros 16 coches proceden de China y los 16 últimos de América.

Paso 9. Se codifica con color naranja a los coches procedentes de China y con morado a los de América.

```
colores2 <- viridis(256)
heatmap(x = datos, scale = "none", col = colores2,
        distfun = function(x){dist(x, method = "euclidean")},
        hclustfun = function(x){hclust(x, method = "average")},
        RowSideColors = rep(c("orange", "purple"), each = 16))
```


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