



**Universidad Autónoma del Estado de Hidalgo**

**Instituto de Ciencias Económico Administrativas**

**Doctorado en Ciencias Económico Administrativas**

**TEMAS SELECTOS I: COMPLEJIDAD ECONÓMICA**

**Laboratorio 37**

**Heatmaps en R con datos no estandarizados**

**Profesor:** Dra. Carla Carolina Pérez Hernández

**Alumna:** Brenda Midhely García Ortiz

Tercer semestre

9 de marzo de 2022.

---

# Heatmaps en R con datos no estandarizados

Midhely García

3/9/2022

CARACTERISTICAS DE LOS AUTOS

```
?mtcars
```

```
mtcars
```

```
##           mpg  cyl  disp  hp drat   wt  qsec vs  am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46 0   1    4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02 0   1    4    4
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61 1   1    4    1
## Hornet 4 Drive  21.4   6 258.0 110 3.08 3.215 19.44 1   0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02 0   0    3    2
## Valiant        18.1   6 225.0 105 2.76 3.460 20.22 1   0    3    1
## Duster 360     14.3   8 360.0 245 3.21 3.570 15.84 0   0    3    4
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00 1   0    4    2
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90 1   0    4    2
## Merc 280       19.2   6 167.6 123 3.92 3.440 18.30 1   0    4    4
## Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90 1   0    4    4
## Merc 450SE     16.4   8 275.8 180 3.07 4.070 17.40 0   0    3    3
## Merc 450SL     17.3   8 275.8 180 3.07 3.730 17.60 0   0    3    3
## Merc 450SLC    15.2   8 275.8 180 3.07 3.780 18.00 0   0    3    3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98 0   0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82 0   0    3    4
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42 0   0    3    4
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47 1   1    4    1
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52 1   1    4    2
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90 1   1    4    1
## Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01 1   0    3    1
## Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87 0   0    3    2
## AMC Javelin    15.2   8 304.0 150 3.15 3.435 17.30 0   0    3    2
## Camaro Z28     13.3   8 350.0 245 3.73 3.840 15.41 0   0    3    4
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05 0   0    3    2
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90 1   1    4    1
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70 0   1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90 1   1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50 0   1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.50 0   1    5    6
## Maserati Bora   15.0   8 301.0 335 3.54 3.570 14.60 0   1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60 1   1    4    2
```

Identificar que tipo de datos tengo

```
class(mtcars)
```

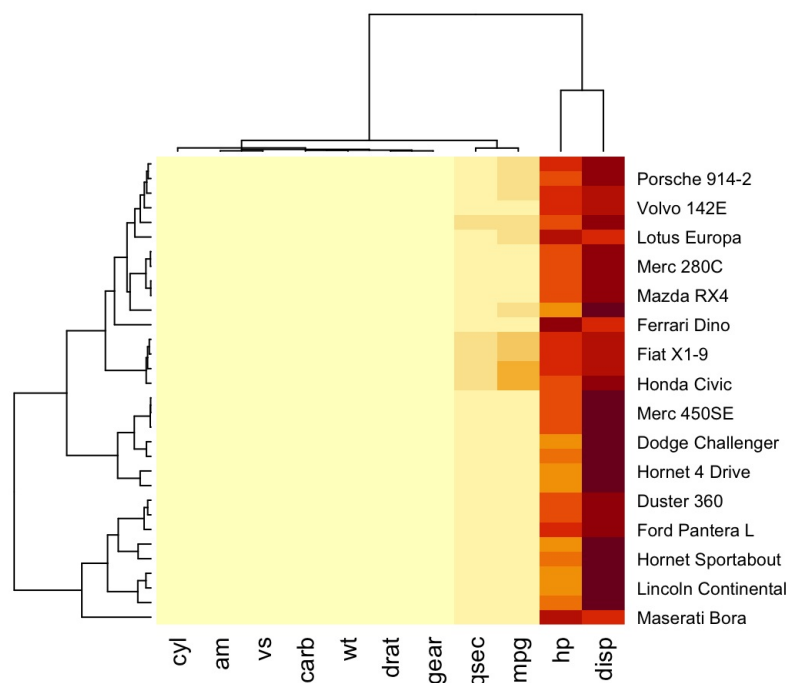
```
## [1] "data.frame"
```

Transformar el data frame en una matriz

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

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

```
heatmap(mtcars_matrix)
```

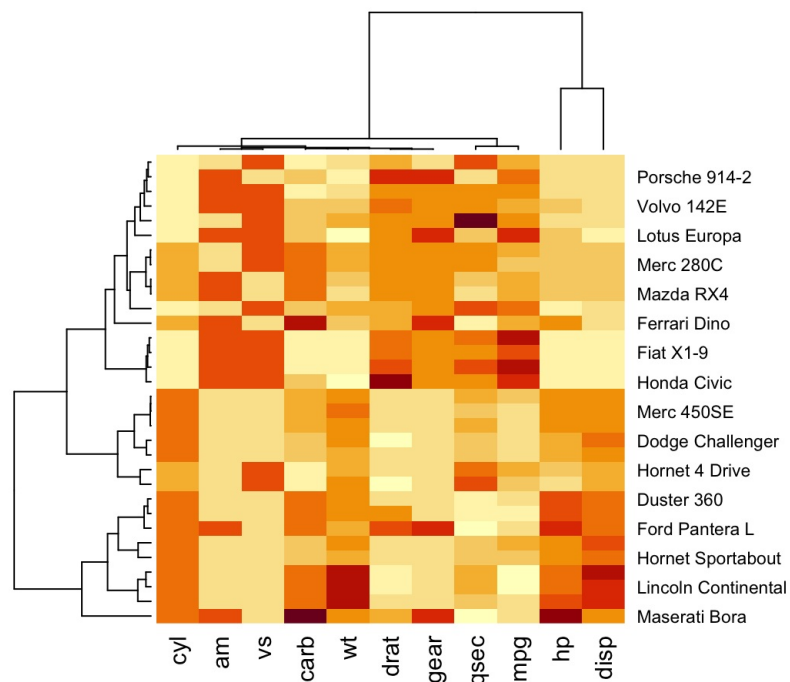


¿Se parece a lo que esperabas? Mire la página de ayuda de la función y lea la descripción del scale argumento en particular

```
?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.

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

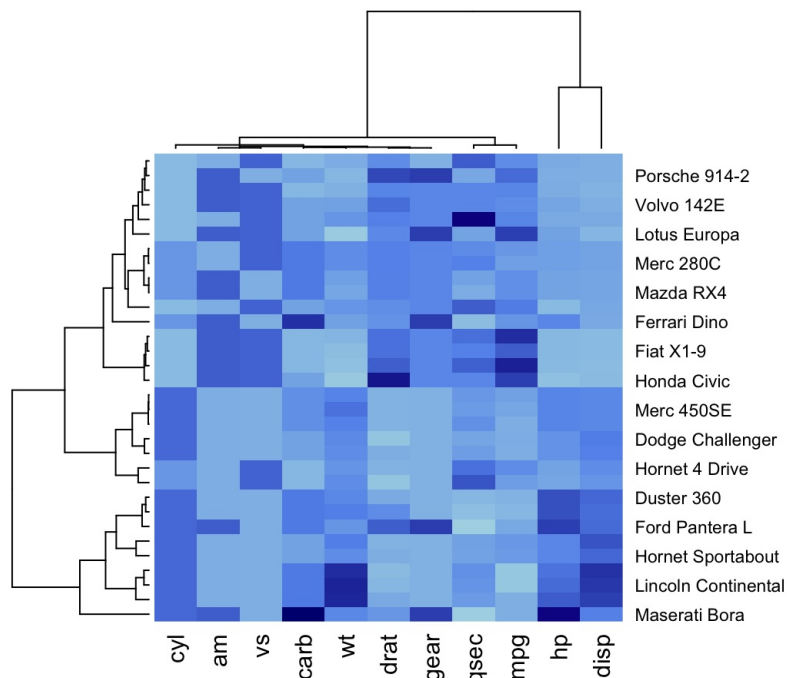


Hacer nuestra propia paleta de colores

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

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

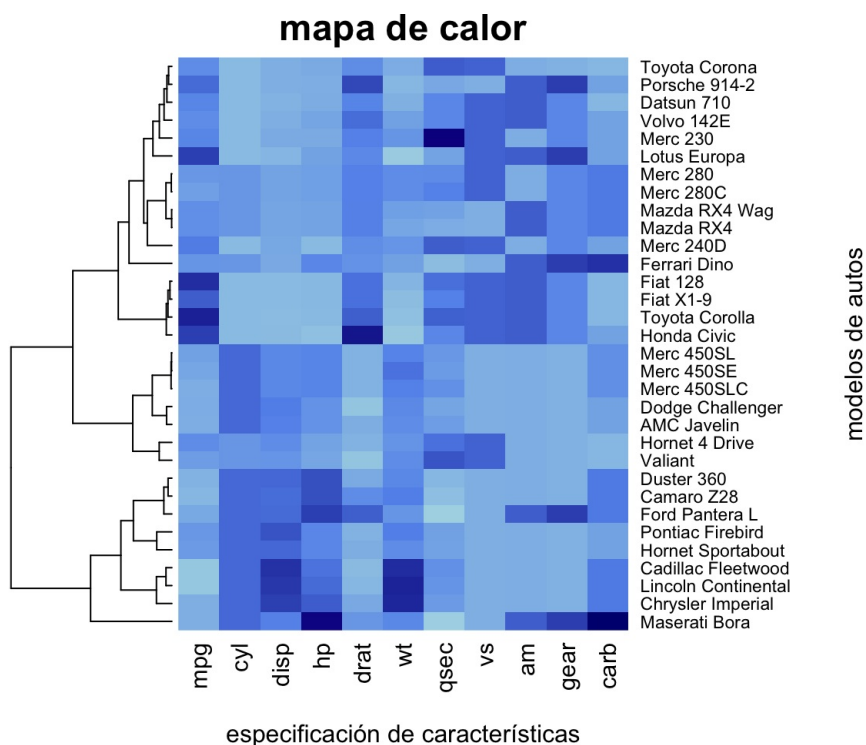
```
heatmap(mtcars_matrix,
  scale = "column",
  col = colores_blue)
```



Eliminar dendrogramas El dendrograma de columna realmente no tiene sentido para este conjunto de datos. Rowv y Colv se puede configurar para NA eliminar dendrogramas, lo que también significa que los datos no se reorganizarán de acuerdo con el método de agrupación.

Se respeta el orden de las columnas, porque se borró el dendrograma de las columnas, ya no se ordena por cluster no se respeta el orden de los renglones, porque estos si tienen clusteo

```
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")
```



```
colnames(mtcars_matrix)
```

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

Asegurarse de tener instalado viridis `install.packages("viridis")` paletas por defecto `rainbow`, `heat.colors`, `terrain.colors`, `topo.colors`, `cm.colors`

Llamar a librería

```
library(viridis)
```

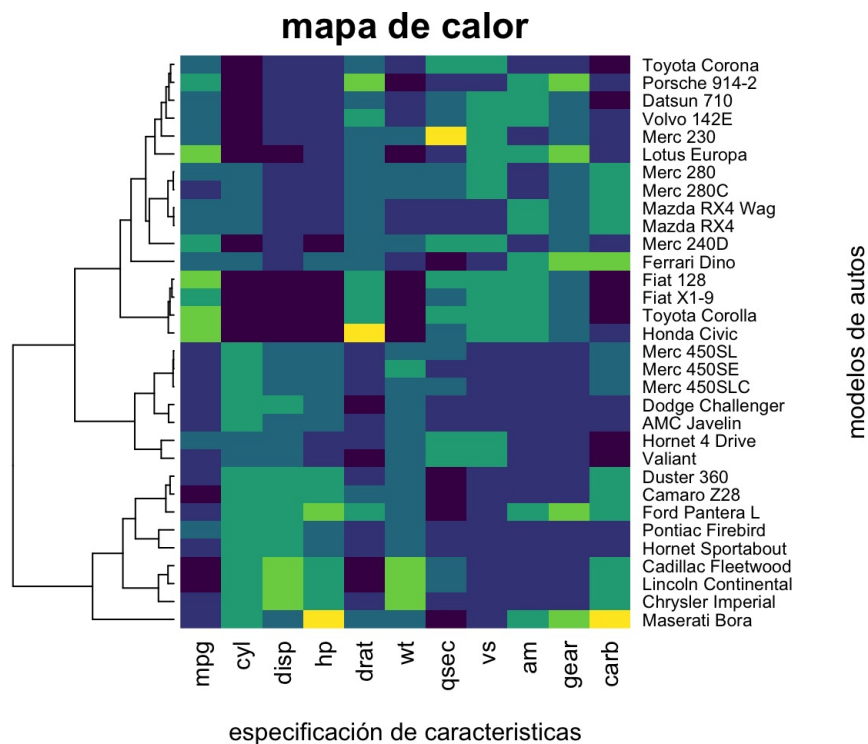
```
## Loading required package: viridisLite
```

```
viridis_pal()
```

```
## function (n)
## {
##   viridisLite::viridis(n, alpha, begin, end, direction, option)
## }
## <bytecode: 0x7fd814390b90>
## <environment: 0x7fd814390458>
```

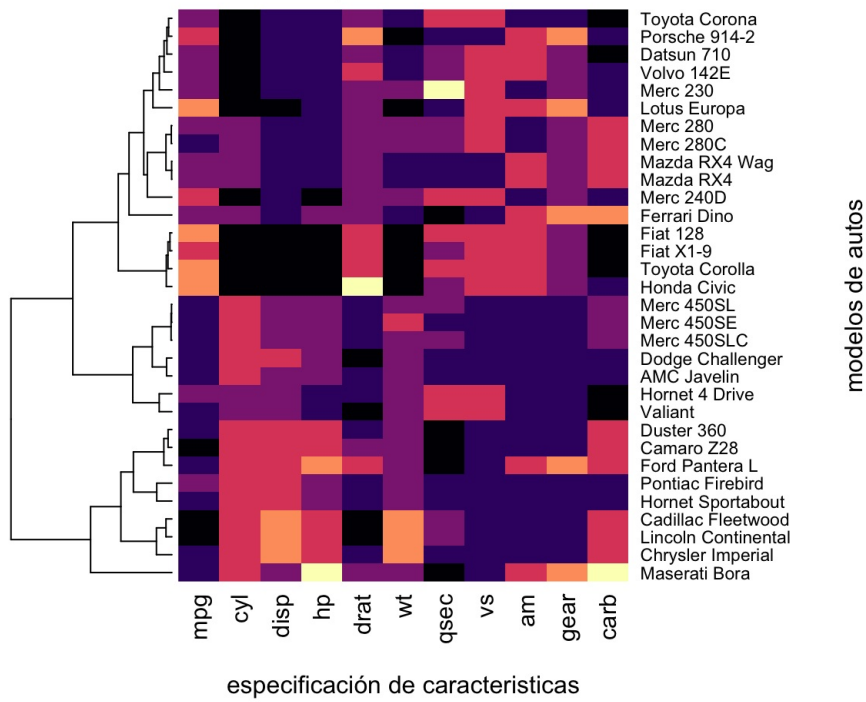
```
col = viridis_pal(option = "viridis")(6)) viridis, magma, plasma, cividis, inferno
```

```
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")
```



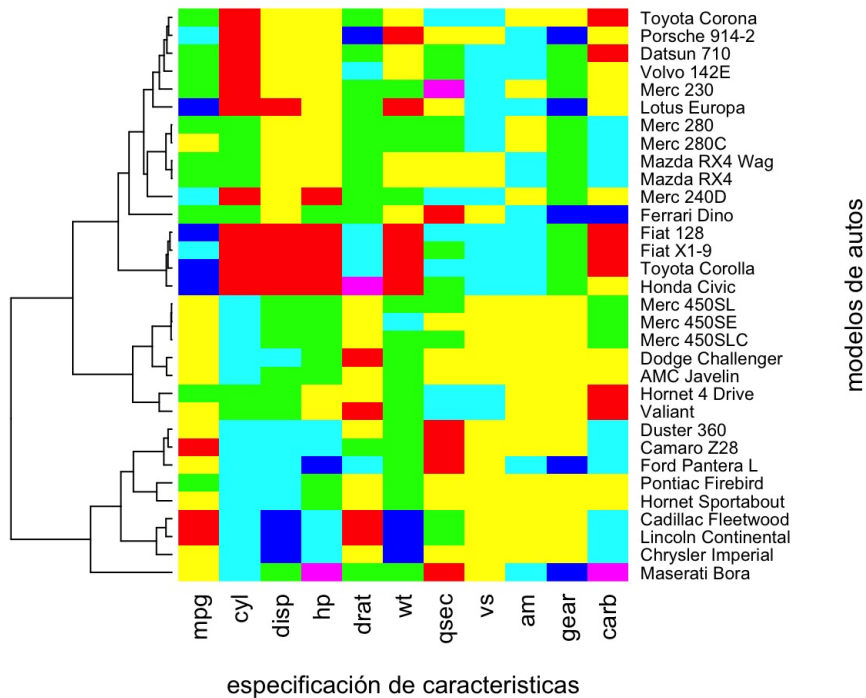
```
heatmap(mtcars_matrix,
  scale = "column",
  col = viridis_pal(option = "magma")(6),
  Colv = NA,
  margins = c(5,10),
  xlab = "especificación de características",
  ylab = "modelos de autos",
  main = "mapa de calor")
```

## mapa de calor



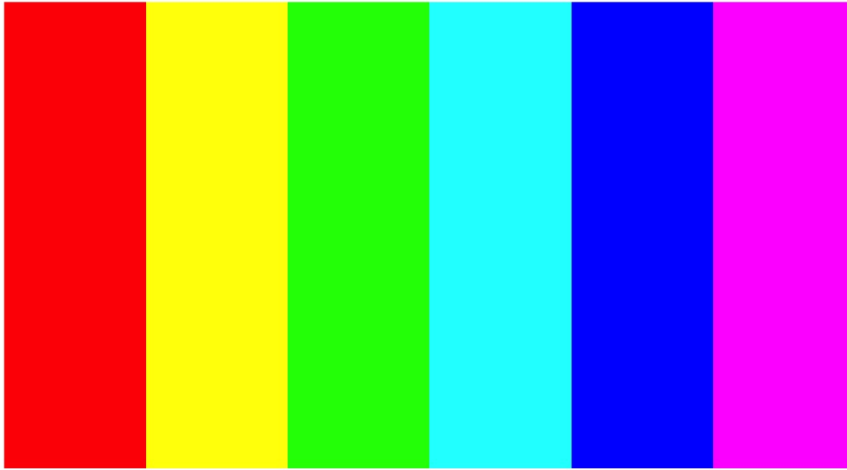
```
heatmap(mtcars_matrix,
  scale = "column",
  col = rainbow (6),
  Colv = NA,
  margins = c(5,10),
  xlab = "especificación de características",
  ylab = "modelos de autos",
  main = "mapa de calor")
```

## mapa de calor



Correr comando para saber valores altos (el valor más bajo es el de la izquierda y el más alto el extremo derecho)

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



Leyenda

Datos extra

```
datos <- mtcars
```

Para que las variables sean comparables bajo un mismo esquema de colores se estandarizan. Técnica para dendograma:

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

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

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

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

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

Paleta de color viridis

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