jerarquicoAglomerativo

September 15, 2021

0.1 Ejercicio

Utilizaremos el dataset *USAarrst* de la libreria cluster. Este conjunto de datos contiene estadísticas, de arrestos por cada 100,000 por asalto, asesinato y violación en cada uno de los 50 estados de EE. UU. En 1973. También se da el porcentaje de la población que vive en áreas urbanas.

Las columnas del datasets:

- **Murder:** arrestos por asesinatos por cada 100,000 hab.
- **Assault:** arrestos por asaltos por cada 100,000 hab.
- **UrbanPop:** porcentaje de población urbana.
- **Rape:** arrestos por violacón por cada 100,000 hab.

Los estados vienen como el nombre de cada fila.

Veremos diferencias entre los métodos jerárquicos que vimos en la teórica y tratar de agrupar estados que tengan comportamientos similares.

[1]:
```
library(cluster)
library(dendextend)
library(ggplot2)
library(factoextra)
library(repr)
```

```
---------------------
Welcome to dendextend version 1.14.0
Type citation('dendextend') for how to cite the package.

Type browseVignettes(package = 'dendextend') for the package vignette.
The github page is: https://github.com/talgalili/dendextend/

Suggestions and bug-reports can be submitted at:
https://github.com/talgalili/dendextend/issues
Or contact: <tal.galili@gmail.com>

        To suppress this message use:
suppressPackageStartupMessages(library(dendextend))
---------------------
```

```
Attaching package: 'dendextend'


The following object is masked from 'package:stats':

    cutree


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

[2]:
```
### Datos

data("USArrests")
df = USArrests
head(df)
```

A data.frame: 6 × 4

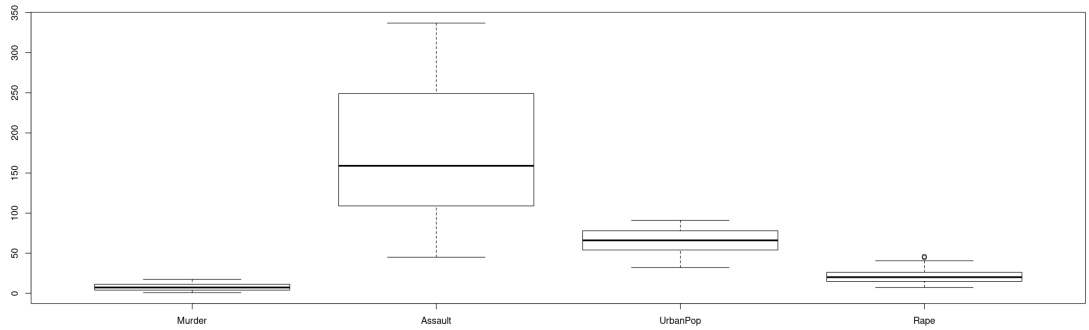
	Murder <dbl>	Assault <int>	UrbanPop <int>	Rape <dbl>
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

0.1.1 Ojo con las escalas!

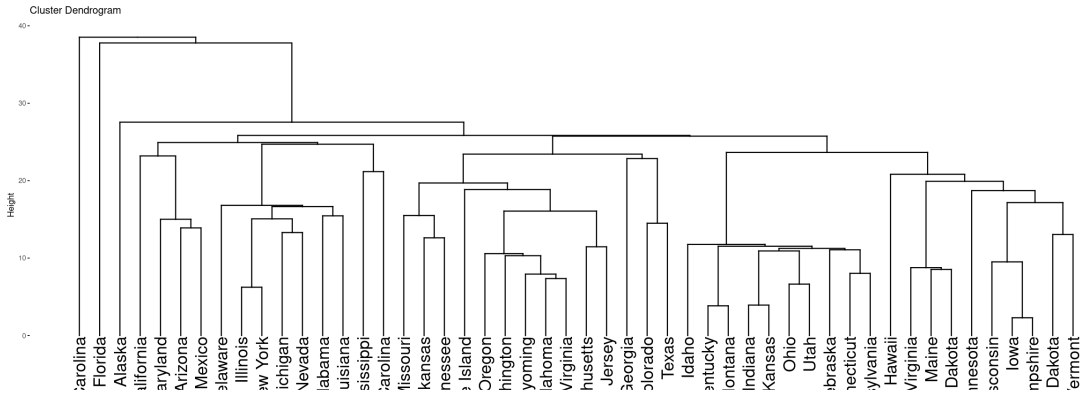
Si las variables tienen escalas muy diferentes, eso puede afectar el análisis de clusters.

[3]:
```
### Veamos las escalas

summary(df)
options(repr.plot.width = 20) #, repr.plot.height = 0.75, repr.plot.res = 100)
boxplot(df)
```

```
     Murder          Assault         UrbanPop          Rape
 Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30
 1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07
 Median : 7.250   Median :159.0   Median :66.00   Median :20.10
 Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23
 3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18
 Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00
```



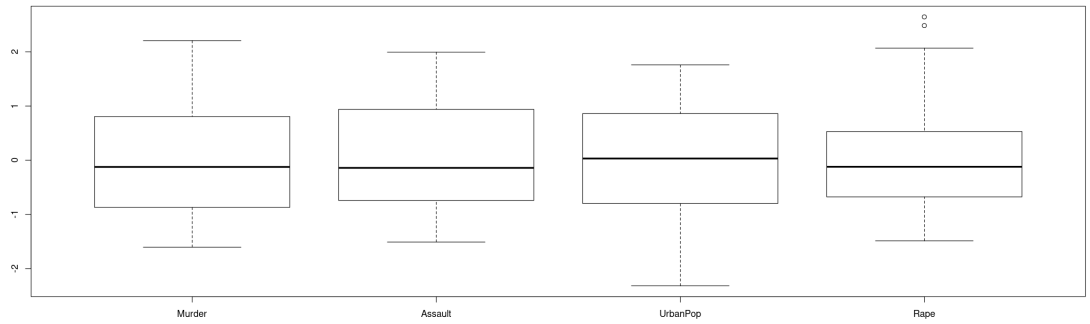
[4]:
```
### Veamos si la escala realmente puede influir en el análisis. Utilicemos el␣
 ↪método single
## Construimos el primer dendrograma

d = dist(df) #Calculamos la matriz de distancias

arrest_hclust1 = hclust(d, method="single")
fviz_dend(arrest_hclust1, cex=1.5)
```



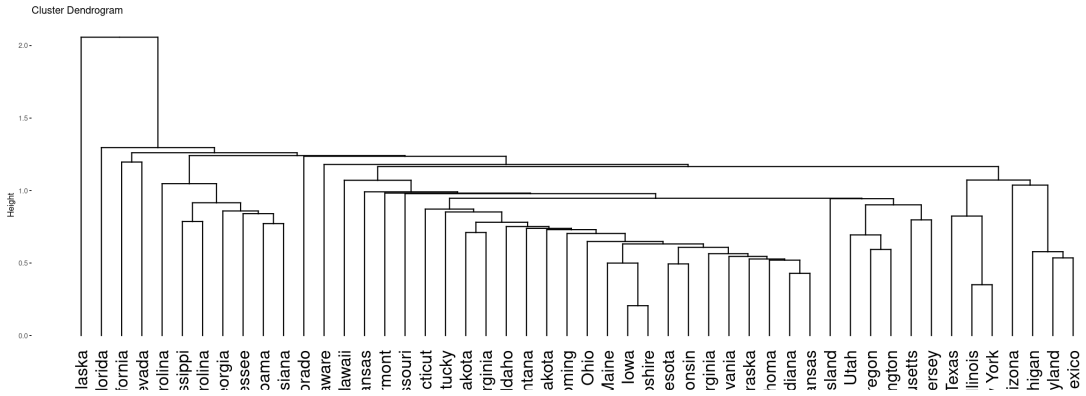
[5]:
```
### Ahora estandaricemos las variables

df_stand = scale(df)
head(df_stand)
boxplot(df_stand)
```

A matrix: 6 × 4 of type dbl

	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

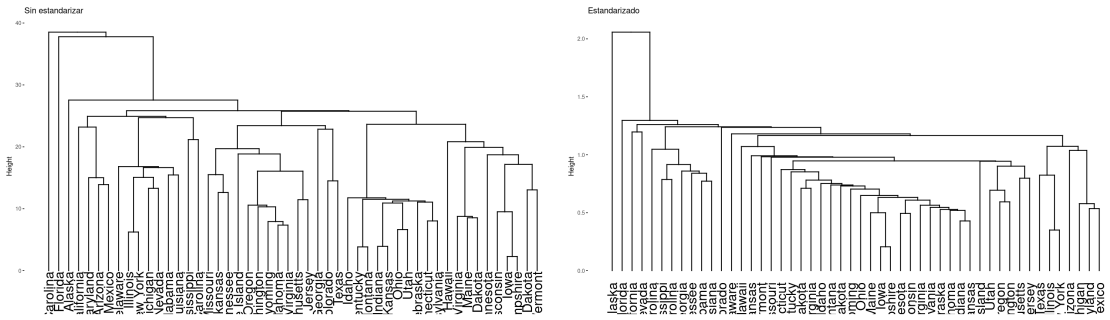


[7]:
```
### Construimos el segundo dendrograma

d_stand = dist(df_stand)
arrest_hclust2 = hclust(d_stand, method="single")
fviz_dend(arrest_hclust2, cex=1.5)
```



[8]:
```
## Comparemos

library(gridExtra)

options(repr.plot.width = 25)
```

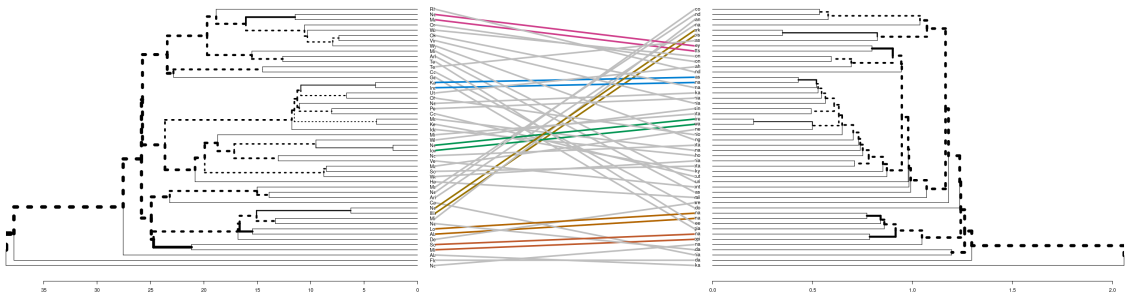
```
s1 = fviz_dend(arrest_hclust1, main="Sin estandarizar", cex=1.5)
s2 = fviz_dend(arrest_hclust2, main="Estandarizado", cex=1.5)
gridExtra::grid.arrange(s1,s2, ncol=2)
```



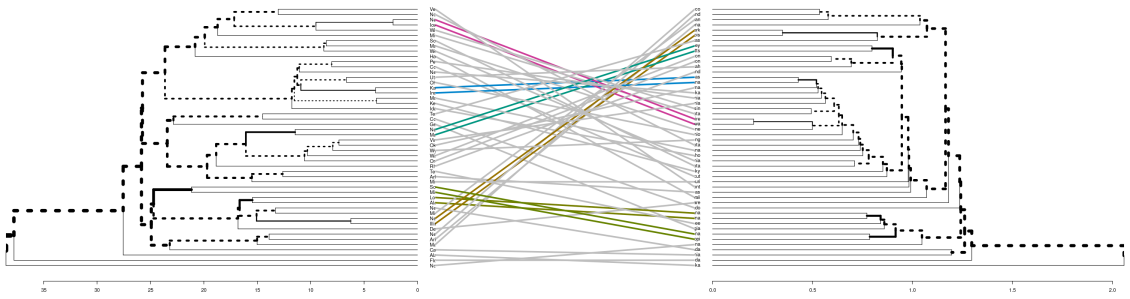
Mini ejercicio: Calcular la disimilaridad utilizando el dataset sin estandarizar entre Alabama y Florida. Ver qué variable aporta más a la disimilaridad. ¿Qué sucede en el caso estandarizado?

[9]:
```
### Veamos otras formas de comparar los dendrogramas

# Creamos los dendrogramas
dend1 = as.dendrogram(arrest_hclust1)
dend2 = as.dendrogram(arrest_hclust2)

# Los juntamos en una lista
dend_list = dendlist(dend1, dend2)
```

[10]:
```
# Alineamos los mejor posible los dendrogramas

dendlist(dend1, dend2) %>%
    untangle(method = "step1side") %>% # Busca la mejor alineación
    tanglegram() # Dibuja
```



[11]:
```
### Observar la diferencia entre alinear y no hacerlo.

dendlist(dend1, dend2) %>%
    tanglegram() # Draw the two dendrograms
```



[12]:
```
# Calculamos una medida de alineamiento entre los dendrogramas

dendlist(dend1, dend2) %>%
    untangle(method = "step1side") %>% #
    entanglement()                     # valor entre 0 y 1. Cuanto más cerca de␣
 ↪1 menos alineados.
```

0.354523114674059

[13]:
```
# Lo mismo sin la alineación

dendlist(dend1, dend2) %>%
    entanglement()
```

0.505215866316951

[14]:
```
# Correlación entre los árboles

dend_list = dendlist(dend1, dend2)

# Cophenetic correlation
cor.dendlist(dend_list, method = "cophenetic")

# Baker correlation
cor.dendlist(dend_list, method = "baker")
```

A matrix: 2 × 2 of type dbl

1.0000000	0.4104309
0.4104309	1.0000000

A matrix: 2 × 2 of type dbl

1.0000000	0.6343639
0.6343639	1.0000000

0.1.2 Moraleja: es importante escalar los datos para que estén todos en el mismo rango.

Ahora comparemos distintos métodos jerárquicos aglomerativos y veamos que tan distintos son los resultas

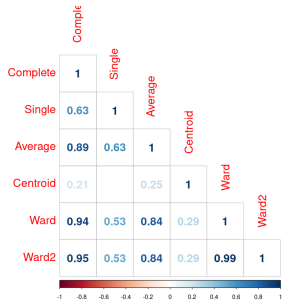
[15]:
```
### Ya nos convencimos que hay que estandarizar
### Analicemos los resultados que nos hubiesen dado distintos métodos

# Create multiple dendrograms by chaining
dend1 <- df_stand %>% dist %>% hclust("complete") %>% as.dendrogram
dend2 <- df_stand %>% dist %>% hclust("single") %>% as.dendrogram
dend3 <- df_stand %>% dist %>% hclust("average") %>% as.dendrogram
dend4 <- df_stand %>% dist %>% hclust("centroid") %>% as.dendrogram
dend5 <- df_stand %>% dist %>% hclust("ward.D") %>% as.dendrogram
dend6 <- df_stand %>% dist %>% hclust("ward.D2") %>% as.dendrogram

# Compute correlation matrix
dend_list <- dendlist("Complete" = dend1, "Single" = dend2,
                      "Average" = dend3, "Centroid" = dend4, "Ward"=dend5,␣
 ↪"Ward2"=dend6)
cors <- cor.dendlist(dend_list, method="baker")

# Print correlation matrix
round(cors, 2)
```

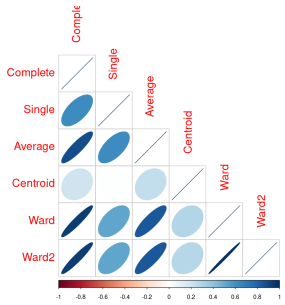
A matrix: 6 × 6 of type dbl

	Complete	Single	Average	Centroid	Ward	Ward2
Complete	1.00	0.63	0.89	0.21	0.94	0.95
Single	0.63	1.00	0.63	0.01	0.53	0.53
Average	0.89	0.63	1.00	0.25	0.84	0.84
Centroid	0.21	0.01	0.25	1.00	0.29	0.29
Ward	0.94	0.53	0.84	0.29	1.00	0.99
Ward2	0.95	0.53	0.84	0.29	0.99	1.00

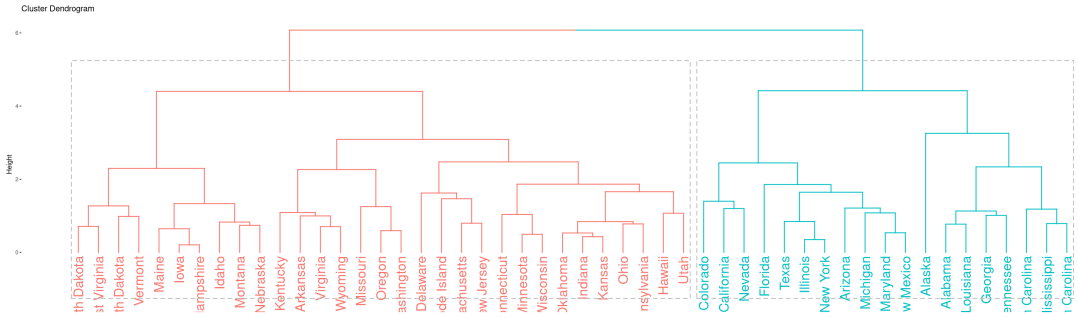
[16]:
```
# Visualize the correlation matrix using corrplot package
library(corrplot)
options(res.plot.width=100)
par(mfrow=c(1,2))
corrplot(cors, "number", "lower", tl.cex=1.5, number.cex=1.5)
corrplot(cors, "ellipse", "lower", tl.cex=1.5)
```

```
corrplot 0.84 loaded
```





[18]:
```
### Tomemos alguno de los métodos que más coinciden: average, complete, ward, ward2.

## Observemos el dendrograma y hagamos un corte

fviz_dend(dend1, k=2, rect=TRUE, cex=1.5)

# Eligiendo a ojo podriamos decir que hay x grupos
```



[19]:
```
## Cortemos el árbol

complete_hclust = hclust(dist(df_stand), method="complete")
complete_clusters = cutree(complete_hclust, k=4)
complete_clusters
```

Alabama 1 **Alaska** 1 **Arizona** 2 **Arkansas** 3 **California** 2 **Colorado** 2 **Connecticut** 3 **Delaware** 3 **Florida** 2 **Georgia** 1 **Hawaii** 3 **Idaho** 4 **Illinois** 2 **Indiana** 3 **Iowa** 4 **Kansas** 3 **Kentucky** 3 **Louisiana** 1 **Maine** 4 **Maryland** 2 **Massachusetts** 3 **Michigan** 2 **Minnesota** 3 **Mississippi** 1 **Missouri** 3 **Montana** 4 **Nebraska** 4 **Nevada** 2 **New Hampshire** 4 **New Jersey** 3 **New Mexico** 2 **New York** 2 **North Carolina** 1 **North Dakota** 4 **Ohio** 3 **Oklahoma** 3 **Oregon** 3 **Pennsylvania** 3 **Rhode Island** 3 **South Carolina** 1 **South Dakota** 4 **Tennessee** 1 **Texas** 2 **Utah** 3 **Vermont** 4 **Virginia** 3 **Washington** 3 **West Virginia** 4 **Wisconsin** 3 **Wyoming** 3

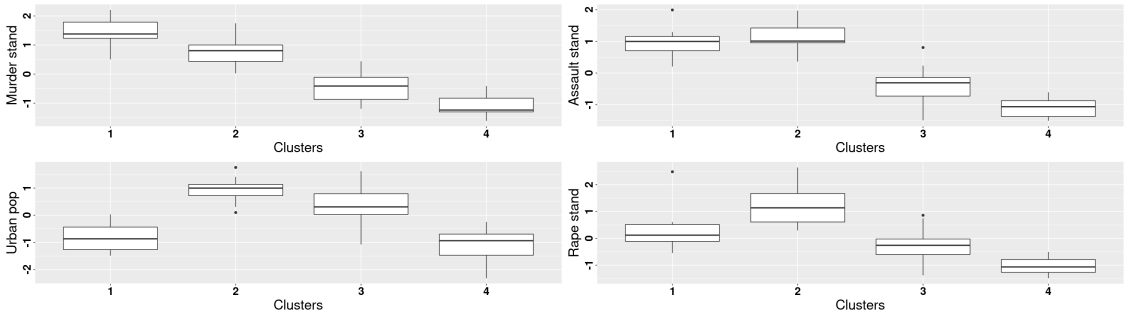
[20]:
```
### Hagamos un ligero análisis de nuestros grupos. ¿Qué característica␣
↪comparten?

df_stand = scale(df)
df_stand2 = data.frame(df_stand, clus=complete_clusters) # Creo un nuevo␣
↪dataframe con la columna para cluster
df_stand2$clus = factor(df_stand2$clus) # convertir la columna que indica el␣
↪numero de cluster de int a factor
head(df_stand2)
```

A data.frame: 6 × 5

	Murder <dbl>	Assault <dbl>	UrbanPop <dbl>	Rape <dbl>	clus <fct>
Alabama	1.24256408	0.7828393	-0.5209066	-0.003416473	1
Alaska	0.50786248	1.1068225	-1.2117642	2.484202941	1
Arizona	0.07163341	1.4788032	0.9989801	1.042878388	2
Arkansas	0.23234938	0.2308680	-1.0735927	-0.184916602	3
California	0.27826823	1.2628144	1.7589234	2.067820292	2
Colorado	0.02571456	0.3988593	0.8608085	1.864967207	2

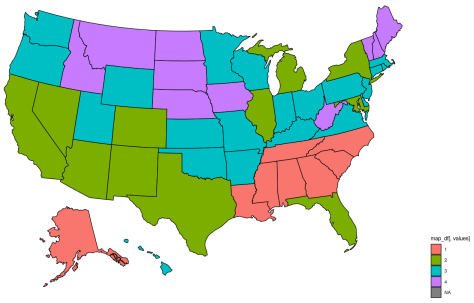
[21]:
```
### Hagamos un ligero análisis de nuestros grupos. ¿Qué característica␣
↪comparten?

bb1 = ggplot(data = df_stand2, aes(x=clus, y=Murder, group=clus)) + ␣
↪geom_boxplot() +
   xlab("Clusters") + ylab("Murder stand") +
   theme(axis.text.x = element_text(face="bold", colour="black", size=rel(2),␣
↪angle=0, hjust=0.5),
         axis.text.y = element_text(face="bold", colour="black", size=rel(2),␣
↪angle=90, hjust=0.5),
         axis.title.x = element_text(size = rel(2)), axis.title.y =␣
↪element_text(size = rel(2))
        )

bb2 = ggplot(data = df_stand2, aes(x=clus, y=Assault, group=clus)) + ␣
↪geom_boxplot() +
   xlab("Clusters") + ylab("Assault stand") +
   theme(axis.text.x = element_text(face="bold", colour="black", size=rel(2),␣
↪angle=0, hjust=0.5),
         axis.text.y = element_text(face="bold", colour="black", size=rel(2),␣
↪angle=90, hjust=0.5),
         axis.title.x = element_text(size = rel(2)), axis.title.y =␣
↪element_text(size = rel(2))
        )

bb3 = ggplot(data = df_stand2, aes(x=clus, y=UrbanPop, group=clus)) + ␣
↪geom_boxplot() +
```

```
    xlab("Clusters") + ylab("Urban pop") +
    theme(axis.text.x = element_text(face="bold", colour="black", size=rel(2),␣
 ↪angle=0, hjust=0.5),
          axis.text.y = element_text(face="bold", colour="black", size=rel(2),␣
 ↪angle=90, hjust=0.5),
          axis.title.x = element_text(size = rel(2)), axis.title.y =␣
 ↪element_text(size = rel(2))
         )

bb4 = ggplot(data = df_stand2, aes(x=clus, y=Rape, group=clus)) + ␣
 ↪geom_boxplot() +
    xlab("Clusters") + ylab("Rape stand") +
    theme(axis.text.x = element_text(face="bold", colour="black", size=rel(2),␣
 ↪angle=0, hjust=0.5),
          axis.text.y = element_text(face="bold", colour="black", size=rel(2),␣
 ↪angle=90, hjust=0.5),
          axis.title.x = element_text(size = rel(2)), axis.title.y =␣
 ↪element_text(size = rel(2))
         )

gridExtra::grid.arrange(bb1,bb2,bb3,bb4, ncol=2)
```



[22]:
```
### Podemos ayudarnos con variables externas

# Voy a considerar la posición espacial de cada cluster en un mapa. No␣
 ↪olvidemos que las observaciones se corresponden con estados en EEUU.

library(usmap)

df_stand3 = data.frame(df_stand2, state=rownames(df_stand))
```

[23]:
```
plot_usmap(data = df_stand3, values="clus", color="black") + theme(legend.
 ↪position = "right")
```



0.2 Ejercicio:

Esto debe hacerlo una vez en su vida, pero debe hacerlo. Aplicar a mano el algoritmo "average linkage" para el siguiente conjunto de datos:

$$D = \{(1, 2), (2, 3.4), (1.1, 7), (-2, -0.5), (2.2, 5), (3.2, 3.1), (-1, -1)\}.$$

Utilizar la distancia Euclídea. La primer matriz de disimilaridad puede hacerla con la compu.

0.3 Ejercicio:

1- Repetir el análisis de clusters utilizando otro método que no sea complete.

2- Repetir el análisis sacando la variable UrbanPop.

0.4 Ejercicio: analizar el conjunto de los "Mall_Customers.csv"

Puede ver la descripción de los datos en kaggle.

Pruebe usar el método divisivo Daisy, comparelo con el agglomerativo que más le guste.

[ ]: