

Format d'entrega

Aquest document s'ha realitzat mitjançant **Markdown**¹ amb l'ajuda de l'entorn de desenvolupament **RStudio**² utilitzant les característiques que aquest ofereix per a la creació de documents **R** reproduïbles.

La documentació generada en la realització de la pràctica es troba allotjada en **GitHub** al següent repositori:

- <https://github.com/rsanchezs/data-minig>

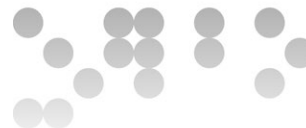
En aquest repositori es poden trobar els següents fitxers:

- Aquest document en formats **pdf** i **docx** amb el nom `rsanchezs_PAC2`.
- Un document **R Markdown**³ que es pot utilitzar per a reproduir tots els exemples presentats a la PAC.
- El conjunt de dades utilitzades.

¹ La documentació oficial es pot trobar a: <http://www.sthda.com/english/rpkgs/factoextra>.

² Per a més informació: <https://es.wikipedia.org/wiki/K-medoids>

³ <https://rmarkdown.rstudio.com/>



1. Nota: Propietat intel·lectual

Sovint és inevitable, al produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-lo en el marc d'una pràctica dels Estudis, sempre que això es documenti clarament i no suposi plagi en la pràctica.

Per tant, al presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en quin es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU, GPL ...).

L'estudiant haurà d'assegurar-se que la llicència no impedeix específicament el seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida per copyright.

Hauríeu de, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.

Exercici 1

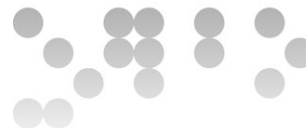
Exercici 2

Requisits

Per començar, per a la realització del nostre anàlisi necessitarem els següents paquets:

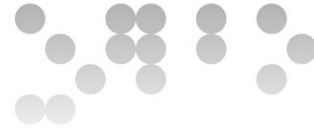
- `cluster` per a la computació dels algorismes d'agregació.
- `factoextra` per a la visualització de resultats d'agregació i que es fonamenta en el paquet `ggplot2`.⁴

⁴ La documentació oficial es pot trobar a: <http://www.sthda.com/english/rpkgs/factoextra>.



El paquet `factoextra` conté funcions per anàlisi de *clustering* i visualització dels resultats:

Funció	Descripció
<code>dist(fviz_dist, get_dist)</code>	Visualització i computació de la matriu de distàncies
<code>get_clust_tendency</code>	Avaluació de la tendència d'agregació
<code>fviz_nbclust(fviz_gap_stat)</code>	Determinació del nombre òptim de clústers
<code>fviz_dend</code>	Visualització de dendrogrames
<code>fviz_cluster</code>	Visualització dels resultats d'agrupament
<code>fviz_mclust</code>	Visualització dels resultats del model d'agrupament
<code>fviz_silhouette</code>	Visualització de la informació de la silueta
<code>hkmeans</code>	K-means jerarquic
<code>eclust</code>	Visualització



de l'anàlisi de agrupament

Podem instal·lar els dos paquets com es mostra en la següent línia de codi:

```
# Instal·lació paquets clustering  
install.packages(c("cluster", "factoextra"))
```

En acabar, ens caldrà carregar les llibreries a la sessió R:

```
# Carreguem les llibreries  
library(cluster)  
library(factoextra)
```

Preparació de les dades

D'entrada, per a realitzar una anàlisi d'agregació en R cal assegurar-se d'unes quantes coses:

- Que les files es corresponen a observacions (individuals) i les columnes a variables.
- Qualsevol valor desconegut en el nostre conjunt de dades ha de ser o bé eliminat o bé substituït per exemple amb el valor de la mitjana o per el valor més freqüent.
- Les dades han de ser estar discretitzades.

Per il·lustrar l'anàlisi d'agregació farem ús del conjunt de dades [USArrests](#), que conté dades estadístiques d'agressions, assassinats i violacions en cada un dels 50 estats d'USA l'any 1973.



```
# Carreguem les dades
data("USArrests")
df <- USArrests
```

En primer lloc, podem eliminar els valors desconeguts en el nostre conjunt de dades com es mostra a continuació:

```
# Eliminem valors desconeguts
df <- na.omit(df)
```

En segon lloc, discretitzarem les nostres dades estandaritzant-les amb l'ajuda de la funció `scale()`:

```
# Estandaritzem les variables
df <- scale(df)
head(df, n = 3)
```

##		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

Determinació del nombre de clústers

Per a determinar el nombre de clústers farem ús de la funció `fviz_nbclust()` del paquet `factoextra` que calcula els mètodes Elbow, Silhouette i Gap.

El prototip de la funció es el següent:

```
fviz_nbclust(x, FUNcluster, method = c("silhouette", "wss",
"gap_stat"))
```



on els arguments són els següents:

- **x**: matriu o data frame.
- **FUNcluster**: una funció d'agregació. Valors possibles: kmeans, pam, clara i hcut.
- **method**: mètode per a determinar el nombre òptim de clústers. Valors possibles: **Elbow**, **Silhouette** i **Gap**

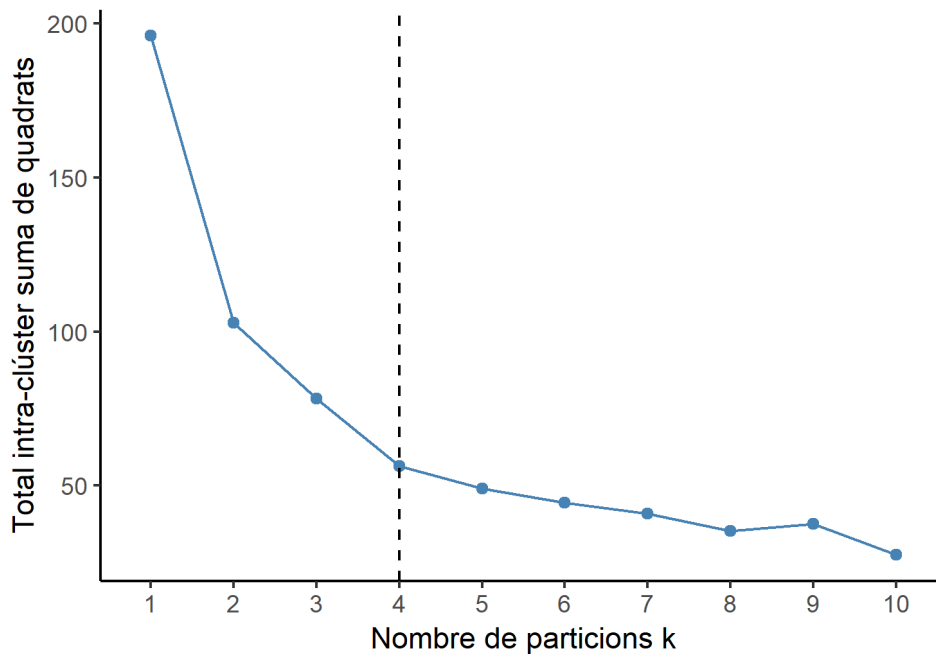
A continuació, es mostra com determinar el nombre òptim de particions per al mètode *k-means*.

```
# Mètode elbow
fviz_nbclust(df, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2) +
  labs(x = "Nombre de particions k",
       y = "Total intra-clúster suma de quadrats",
       title = "Nombre òptim de particions",
       subtitle = "Mètode Elbow") +
  theme_classic() +
  theme(plot.title = element_text(color="#4F81BD", size=14,
face="bold"),
        plot.subtitle = element_text(color="#4F81BD", size=14))
```



Nombre òptim de particions

Mètode Elbow

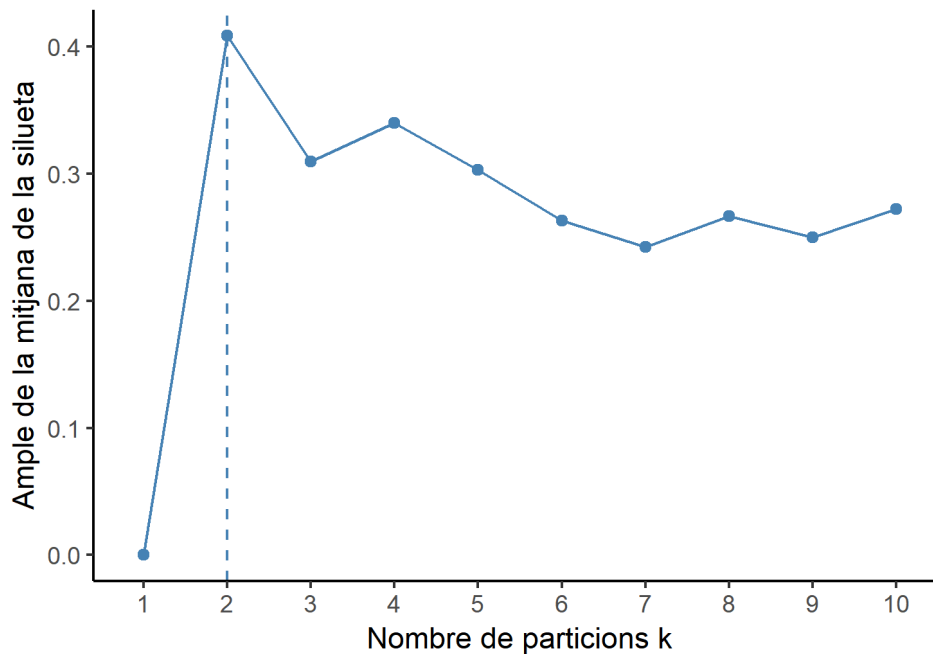


```
# Mètode Silhouette
fviz_nbclust(df, kmeans, method = "silhouette") +
  labs(x = "Nombre de particions k",
       y = "Ample de la mitjana de la silueta",
       title = "Nombre òptim de particions",
       subtitle = "Mètode Silhouette") +
  theme_classic() +
  theme(plot.title = element_text(color="#4F81BD", size=14,
face="bold"),
        plot.subtitle = element_text(color="#4F81BD", size=14))
```

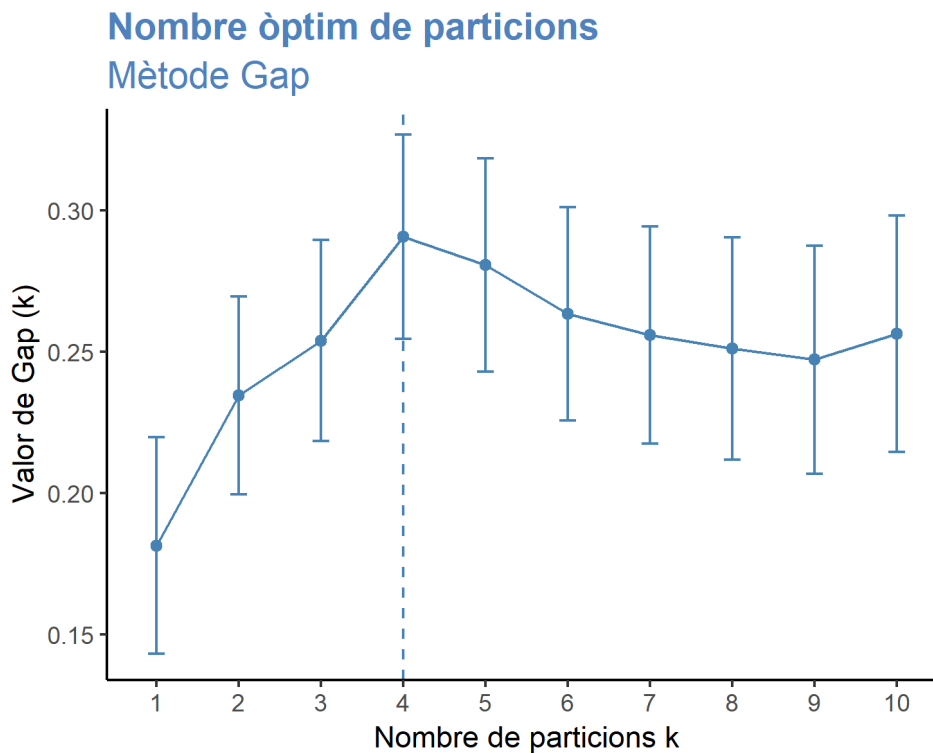
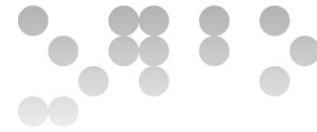


Nombre òptim de particions

Mètode Silhouette



```
# Mètode Gap
set.seed(123)
fviz_nbclust(df, kmeans, nstart = 25,
              method = "gap_stat", nboot = 500) +
  labs(x = "Nombre de particions k",
       y = "Valor de Gap (k)",
       title = "Nombre òptim de particions",
       subtitle = "Mètode Gap") +
  theme_classic() +
  theme(plot.title = element_text(color="#4F81BD", size=14,
face="bold"),
        plot.subtitle = element_text(color="#4F81BD", size=14))
```

Com podem observar en els gràfics:

- El mètode Elbow ens suggereix 4 clústers.
- El mètode Silhoutte ens suggereix 2 clústers.
- El mètode Gap ens sugereix 4 clústers.

Així és que, segons aquestes observacions podem considerar $k = 4$ com el nombre òptim de clústers.

Mètode d'agregació *k-means*

A causa de que, l'algoritme *k-means* comença seleccionant un centroid aleatoriament, es recomanable fer ús de la funció `set.seed()` a l'efecte de



conseguir resultats reproduïbles. Així el lector d'aquest document obtindrà els mateixos resultats que es presenten tot seguit.

A continuació es mostra com aplicar l'algorisme k-means amb $k = 4$:

```
# Execució k-means amb k = 4
set.seed(123)
kmeansFit <- kmeans(df, 4, nstart = 25)
```

Podem mostrar per pantalla els resultats amb la següent línia de codi:

```
# Mostrem els resultats
print(kmeansFit)

## K-means clustering with 4 clusters of sizes 13, 16, 13, 8
##
## Cluster means:
##      Murder      Assault      UrbanPop      Rape
## 1 -0.9615407 -1.1066010 -0.9301069 -0.96676331
## 2 -0.4894375 -0.3826001  0.5758298 -0.26165379
## 3  0.6950701  1.0394414  0.7226370  1.27693964
## 4  1.4118898  0.8743346 -0.8145211  0.01927104
##
## Clustering vector:
##      Alabama      Alaska      Arizona      Arkansas
California
##           4           3           3           4
3
##      Colorado      Connecticut      Delaware      Florida
Georgia
##           3           2           2           3
4
##      Hawaii      Idaho      Illinois      Indiana
Iowa
##           2           1           3           2
1
##      Kansas      Kentucky      Louisiana      Maine
```



```

Maryland
##          2          1          4          1
3
##  Massachusetts      Michigan      Minnesota      Mississippi
Missouri
##          2          3          1          4
3
##      Montana      Nebraska      Nevada      New Hampshire
New Jersey
##          1          1          3          1
2
##      New Mexico      New York      North Carolina      North Dakota
Ohio
##          3          3          4          1
2
##      Oklahoma      Oregon      Pennsylvania      Rhode Island
South Carolina
##          2          2          2          2
4
##      South Dakota      Tennessee      Texas          Utah
Vermont
##          1          4          3          2
1
##      Virginia      Washington      West Virginia      Wisconsin
Wyoming
##          2          2          1          1
2
##
## Within cluster sum of squares by cluster:
## [1] 11.952463 16.212213 19.922437  8.316061
## (between_SS / total_SS =  71.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"

```

Podem observar en la sortida el següent:



- La mitjana de clústers: una matriu, on les files són el nombre de clúster i les columnes són les variables.
- El vector de particions: un vector d'enters (de 1:k) que indica el clúster on cada observació ha sigut agrupada.

Així mateix, és recomanable realitzar un gràfic amb els resultats del model. Ja sigui, per a escollir el nombre de clústers, ja sigui per a comparar diferents anàlisis.

Una possible opció és visualitzar les dades en un diagrama de dispersióicolorint cada observació d'acord al grup assignat.

El problema és que el nostre conjunt de dades conté més de 2 variables i no és possible representar el model en dues dimensions.

Una possible solució és reduir la dimensionalitat fent ús d'un algoritme de reducció del nombre d'atributs, com per exemple **Principal Component Analysis (PCA)**.

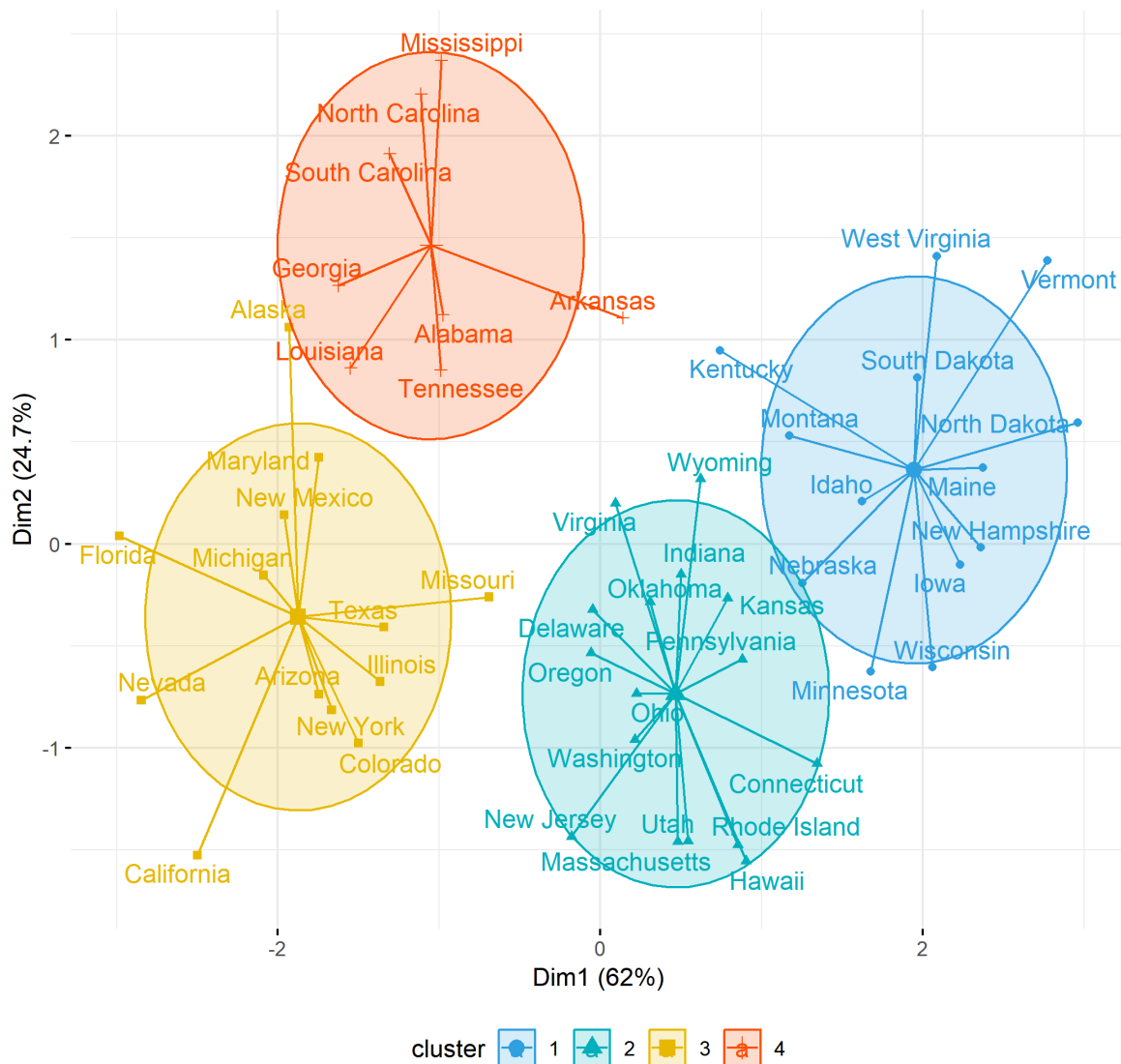
En aquest sentit, farem ús de la funció `fviz_cluster()` que ens permetrà visualitzar els clústers i que utilitza PCA quan el nombre de variables és més gran de 2. Passarem com a arguments el resultat del model i el conjunt de dades original:

```
# Visualitzem els clústers
fviz_cluster(kmeansFit, data = df,
  main = "Gràfic de clústers",
  palette = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),
  ellipse.type = "euclid",
  star.plot = TRUE,
  repel = TRUE,
  ggtheme = theme_minimal()) +
theme(legend.position = "bottom",
```



```
plot.title = element_text(color="#4F81BD", size=16,  
face="bold"))
```

Gràfic de clústers



Podem observar en el gràfic que les observacions són representades mitjançant punts i que en el nostre cas s'ha usat PCA. A més, s'han dibuixat el.lipses per tal de diferenciar cada clúster.



Exercici 3

Mètode d'agregació *k-medoids*

El següent apartat tracta del mètode d'agregació **k-medoids** i la seva implementació mitjançant l'algoritme de **Partició al voltant de Medoids PAM**).

Igual com k-means, K-medoid és una tècnica clàssica de partició de grups que divideix les dades conformades per n objectes en k grups (amb k fixat *a priori*).

És més robust davant el soroll i valors atípics que k-means perquè minimitza una suma de disimilituds (entre parells de punts) en comptes d'una suma de distàncies euclidianes quadrades.

Un **medoid** pot ser definit com l'objecte d'un grup on la seva disimilitud mitjana a tots els objectes en el grup és mínima. És el punt situat més cap al centre en tot el grup.⁵

Per a determinar el nombre de clústers farem ús de la funció `pam()` del paquet `cluster`.

El prototip de la funció es el següent:

```
pam(x, k, metric = "euclidean", stand = FALSE)
```

on els arguments són els següents:

⁵ Per a més informació: <https://es.wikipedia.org/wiki/K-medoids>

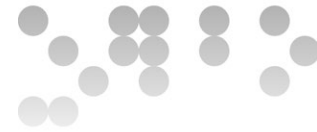


- **x**: on x pot ser:
- Una matriu o data frame de tipus numèric: cada fila correspon a una observació i cada columna a una variable.
- Una matriu de disimilituds: en aquest cas x es normalment la sortida o bé de la funció `daisy()` o bé de `dist()`.
- **K**: el nombre de clusters.
- **metric**: la mesura de similitud. O bé "euclidean" o bé "manhattan".
- **stand**: un valor de tipus lògic; si es TRUE, les variables en x són estandaritzades.

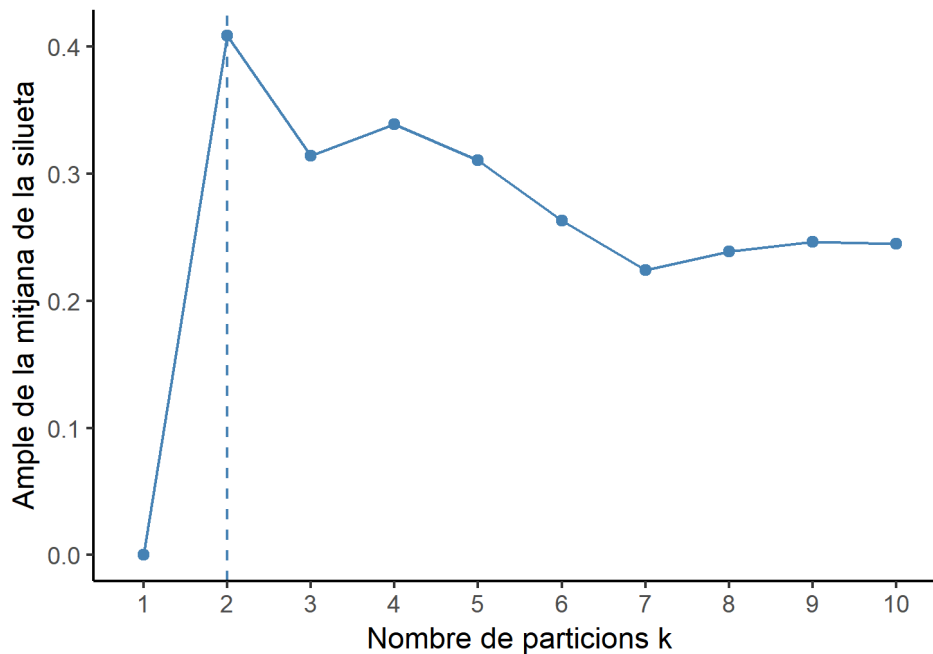
Estimació del nombre òptim de clùsters

Per a estimar el nombre òptim de clùsters utilitzarem la mètrica de Silhoutte. La idea central és calcular l'algorithm PAM amb diferents valors de k. Per a la realització d'aquesta tasca farem ús de la funció `fviz_nbclust()`:

```
# Obtenció de k utilitzant la mètrica Silhoutte
fviz_nbclust(df, pam, method = "silhouette") +
  labs(x = "Nombre de particions k",
       y = "Ample de la mitjana de la silueta",
       title = "Nombre òptim de particions",
       subtitle = "Mètode Silhouette") +
  theme_classic() +
  theme(plot.title = element_text(color="#4F81BD", size=14,
face="bold"),
        plot.subtitle = element_text(color="#4F81BD", size=14))
```



Nombre òptim de particions Mètode Silhouette



Segons el gràfic, el nombre òptim de clústers és 2. En la següent secció, agruparem els objectes en 2 clústers.

Càlcul del mètode PAM

El següent codi calcula el mètode PAM amb $k = 2$:

```
# Execució de l'algoritme PAM
pamFit <- pam(df, 2)
# Visualització de resultats
print(pamFit)
```

```
## Medoids:
```

```
##           ID      Murder      Assault      UrbanPop      Rape
## New Mexico 31  0.8292944  1.3708088  0.3081225  1.1603196
```




```
## Nebraska 27 -0.8008247 -0.8250772 -0.2445636 -0.5052109
## Clustering vector:
## Alabama Alaska Arizona Arkansas
California
## 1 1 1 2
1
## Colorado Connecticut Delaware Florida
Georgia
## 1 2 2 1
1
## Hawaii Idaho Illinois Indiana
Iowa
## 2 2 1 2
2
## Kansas Kentucky Louisiana Maine
Maryland
## 2 2 1 2
1
## Massachusetts Michigan Minnesota Mississippi
Missouri
## 2 1 2 1
1
## Montana Nebraska Nevada New Hampshire
New Jersey
## 2 2 1 2
2
## New Mexico New York North Carolina North Dakota
Ohio
## 1 1 1 2
2
## Oklahoma Oregon Pennsylvania Rhode Island
South Carolina
## 2 2 2 2
1
## South Dakota Tennessee Texas Utah
Vermont
## 2 1 1 2
2
## Virginia Washington West Virginia Wisconsin
Wyoming
## 2 2 2 2
```



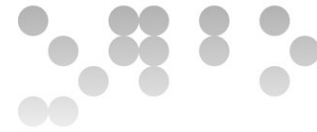
```
2
## Objective function:
##   build      swap
## 1.441358 1.368969
##
## Available components:
## [1] "medoids"      "id.med"      "clustering" "objective"
##      "isolation"
## [6] "clusinfo"     "silinfo"     "diss"       "call"       "data"
```

Podem observar en la sortida el següent:

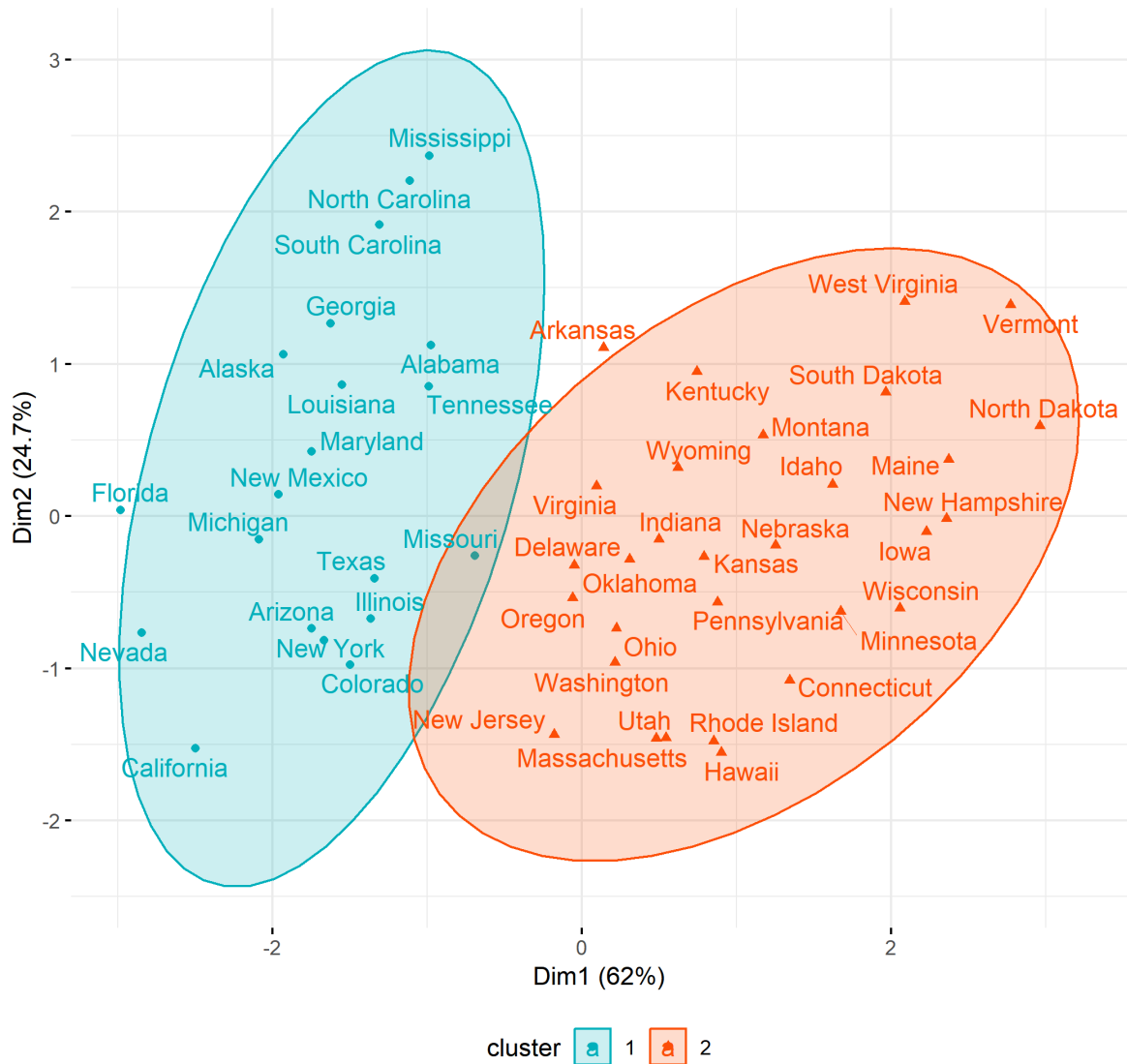
- Els grups medoids: una matriu, on les files són els medoids i les columnes són les variables.
- El vector de particions: un vector d'enters (de 1:k) que indica el clúster on cada observació ha sigut agrupada.

Igual com hem fet en l'exercici anterior utilitzarem la funció `fviz_cluster()` del paquet `factoextra` per a visualitzar les particions:

```
# Visualitzem els clústers
fviz_cluster(pamFit,
  main = "Gràfic de clústers",
  palette = c("#00AFBB", "#FC4E07"),
  ellipse.type = "t",
  repel = TRUE,
  ggtheme = theme_minimal() +
  theme(legend.position = "bottom",
    plot.title = element_text(color="#4F81BD", size=16,
    face="bold"))
```

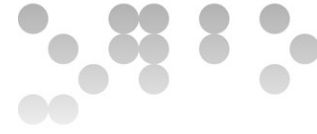


Gràfic de clústers



Bibliografia

[1] Daniel T. Larouse, Chantal D. Larouse: Data Mininig and Predictive Analytics.USA, John Wiley & Sons,2015,ISBN 978-1-118-11619-7



[2] Jordi Gironés Roig, Jordi Casas Roma, Julià Minguillón Alfonso, Ramon Caihuelas Quiles : Minería de Datos: Modelos y Algoritmos. Barcelona, Editorial UOC, 2017, ISBN: 978-84-9116-904-8.

[3] Jiawe Han, Michellie Chamber & Jian Pei: Data mining : concepts and techniques. 3º Edition. USA, Editorial Elsevier, 2012, ISBN 978-0-12-381479-1