# 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**

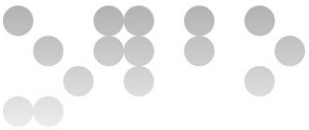
En aquest repositori es poden trobar els següents fitxers:

- 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] **https://www.rstudio.com/**

[3] **https://rmarkdown.rstudio.com/**



# Exercici 1

# Exercici 2

## Requisits

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

- `cluster` per a la computació dels algoritmes d´agregació.
- `factoextra` per a la visualitació de resultats d´agregació i que es fonamenta en el paquet `ggplot2`.[4]

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

| Funció | Descripció |
|---|---|
| `dist(fviz_dist, get_dist)` | Visualització i computació de la matriu de distàncies |
| `get_clust_tendency` | Avaluació de la tendencia d´agregació |
| `fviz_nbclust(fviz_gap_stat)` | Determinació del nombre òptim de clústers |
| `fviz_dend` | Visualització de dendrogrames |
| `fviz_cluster` | Visualització dels resultats d´agrupament |
| `fviz_mclust` | Visualització dels resultats del model d´agrupament |
| `fviz_silhouette` | Visualització de la informació de la silueta |
| `hkmeans` | K-means jerarquic |

[4] La documentació oficial es pot trobar a: **http://www.sthda.com/english/** rpkgs/factoextra.

`eclust` Visualització de l´anàlisi de agrupament

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

```
# Instalació paquets clustering
install.packages(c("cluster", "factoextra"))
```

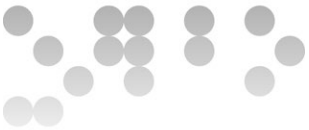
En acabat, ens caldrà carregar les llibreries a la sessió R:

```
# 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.

```
data("USArrests")
df <- USArrests
```

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

```
# Eliminem valor 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
"))
```

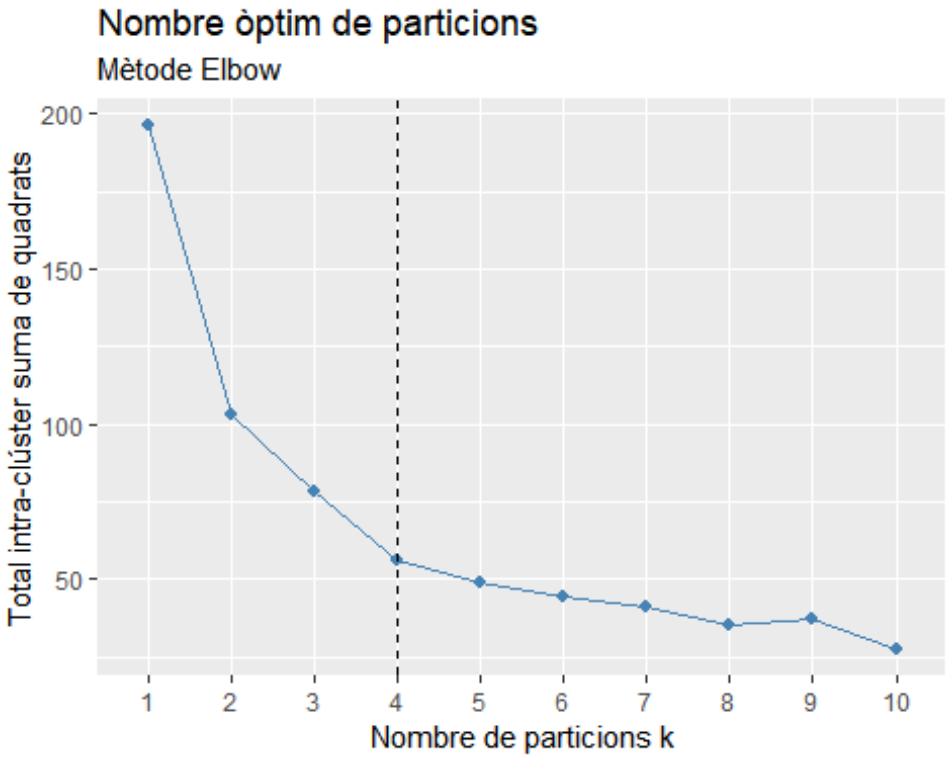
onels arguments són els següents:

- **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**

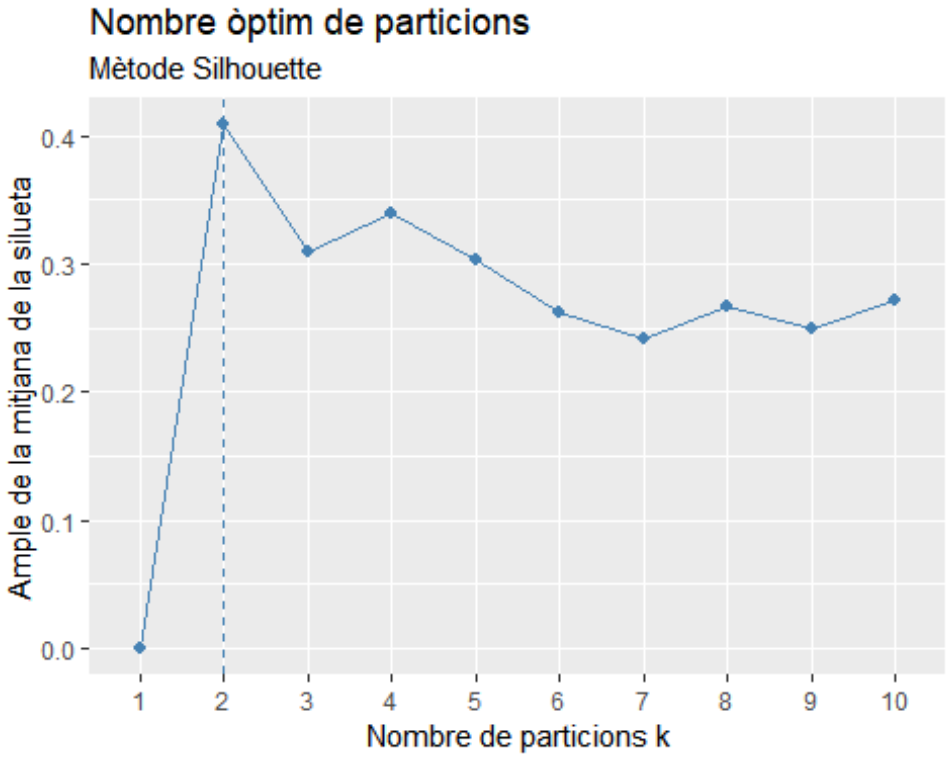
A continuació, es mostra com determinar el nombre òptim de particions per al mètode *k-means*:

```
# 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 qu
```

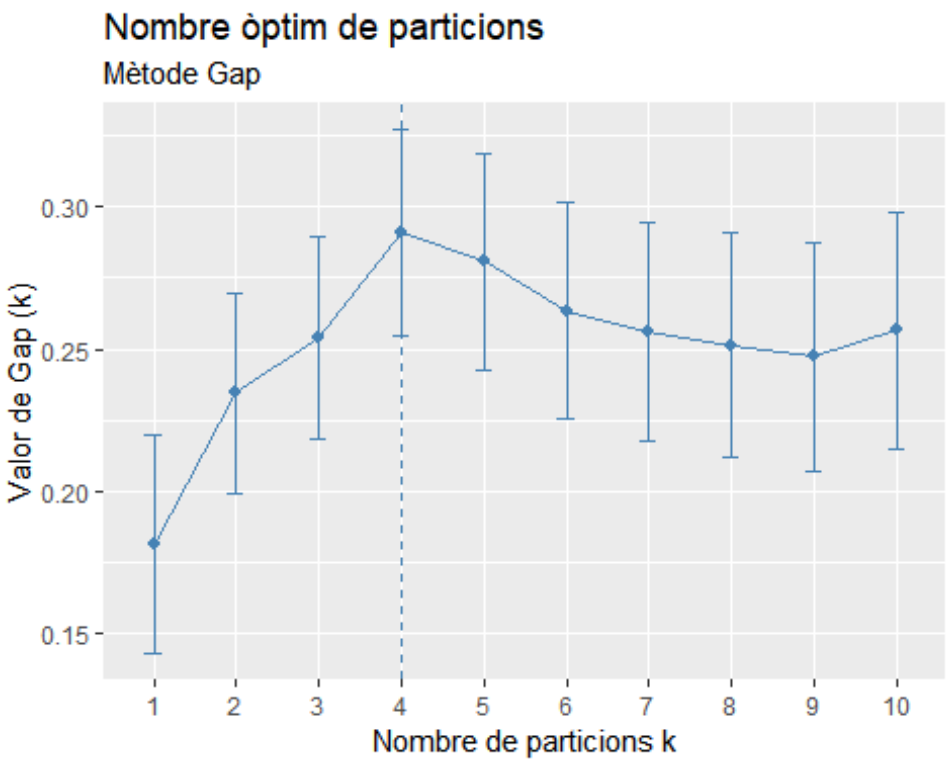
```
adrats",
       title = "Nombre òptim de particions",
       subtitle = "Mètode Elbow") +
  theme_gray()
```




```
# Mètode Silhouette
fviz_nbclust(df, kmeans, method = "silhouette")+
  labs(x = "Nombre de particions k", y = "Ample de la mitjana de la silu
eta",
       title = "Nombre òptim de particions",
       subtitle = "Mètode Silhouette") +
  theme_gray()
```




```
# Mètode Gap
set.seed(123)
fviz_nbclust(df, kmeans, nstart = 25, method = "gap_stat", nboot = 50
0)+
  labs(x = "Nombre de particions k", y = "Valor de Gap (k)",
       title = "Nombre òptim de particions",
       subtitle = "Mètode Gap") +
  theme_gray()
```




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 centroide aleatoriament, es recomanable fer ús de la funció `set.seed()` a l´efecte de conseguir resultats reproduibles. Així el lector d´aquest document obtindrà els mateixos resultats que es presenten tot seguit.

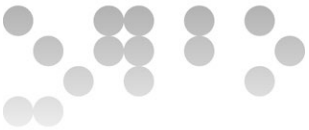
A continuació es mostra com aplicar l´algorisme k-means amb k = 4:

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

Podem mostrar per pantalla els resultats amb la següent línea 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 Sou
th 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.

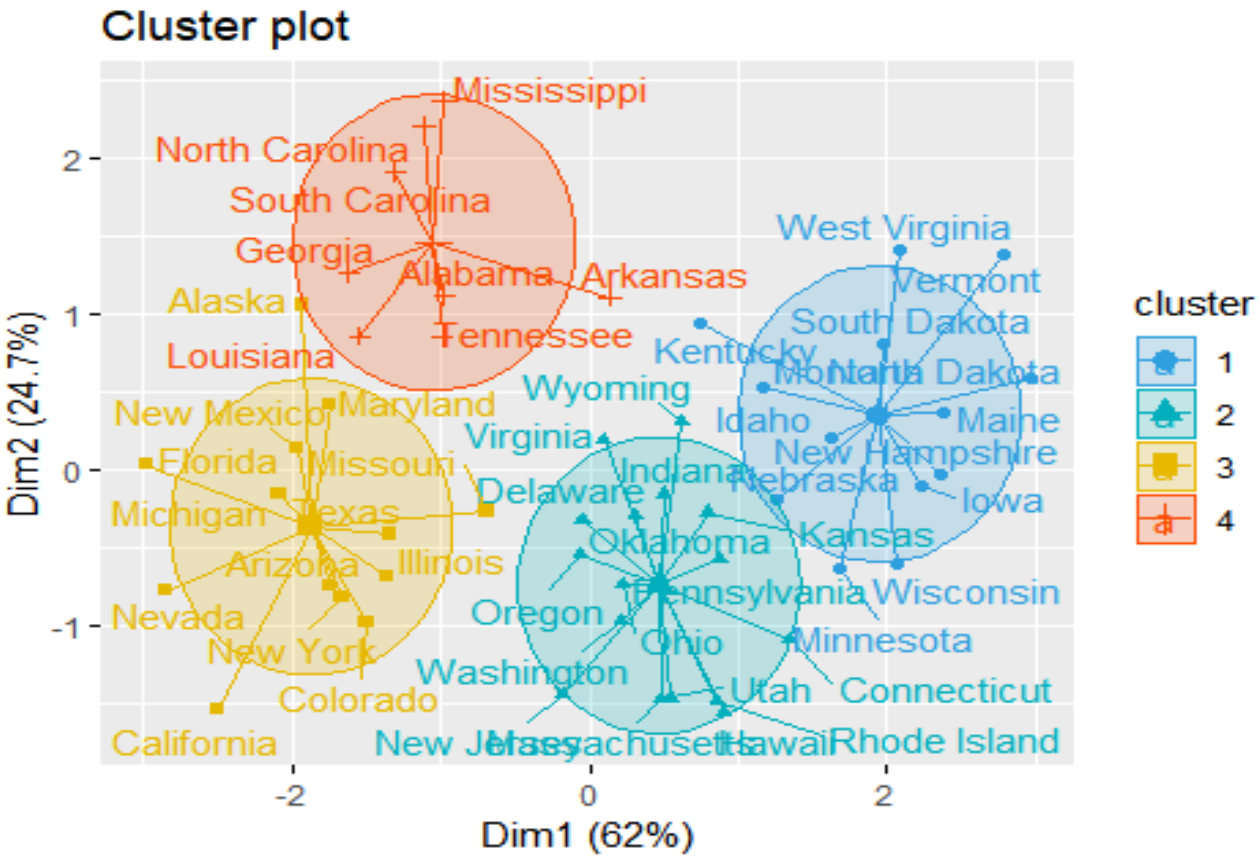
Una possible opció és visualitzar les dades en un diagrama de dispersió acolorint cada observació d'acord al grup assignat.

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

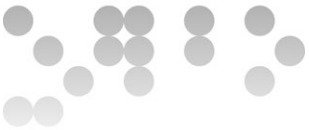
Una possible solució és reduir la dimensionalitat fent ús d´un algoritme de reducció del nombre d´atributs, com per exemple **Principal Component Analysis (PCA)**.

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 els resultats del model i el conjunt de dades original:

```
# Visualitzem els clústers
fviz_cluster(kmeansFit, data = df,
  palette = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),
  ellipse.type = "euclid", # Agrupacions en elipses
  star.plot = TRUE, # Afegeix rectes des de els centroides a les obs
ervacions
  repel = TRUE,
  ggtheme = theme_gray()
)
```




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.



# 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