Tipologia i cicle de vida de les dades: PRA2

Autor: Vicenç Pio i Begoña Felip

Maig 2021

Contents

Tipologia i cicle de vida de les dades

Exercici 1:

Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre?

Font de les dades: Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic)

L'enfonsament del RMS Titanic és un dels naufragis més tràgics de la història. El 15 d'abril de 1912, durant el seu viatge inaugural, el Titanic es va enfonsar després de xocar amb un iceberg i va matar 1502 de 2224 passatgers i tripulants. Aquesta catàstrofe va impactar la comunitat internacional i va conduir a una millor normativa de seguretat per als vaixells. Un dels motius pels quals el naufragi va provocar tanta pèrdua de vides va ser que no hi havia prou vaixells salvavides per als passatgers i la tripulació. Tot i que hi va haver algun element de sort per sobreviure a l'enfonsament, alguns grups de persones tenien més probabilitats de sobreviure que d'altres, com ara dones, nens i la classe alta. La pregunta seria analitzar quin tipus de passatgers tenien més probabilitat de sobreviure. S'aplicaran les eines d'aprenentatge automàtic per predir quins passatgers sobreviurien a la tragèdia.

Disposem de dos grups de dades:

Conjunt d'entrenament (train.csv). Aquest conjunt és el que s'utilitza per a construir el model d'aprenentatge automàtic.

Conjunt de proves (test.csv). Aquest conjunt s'utilitzarà per veure el rendiment del model en dades les quals no disposem. Per a cada passatger del conjunt de proves, s'utilitza el model que prèviament s'ha entrenat per predir si el passatger va sobreviure o no a l'enfonsament del Titanic.

Exercici 2:

Integració i selecció de les dades d'interès a analitzar.

```
trainData <- read.csv('data/train.csv',stringsAsFactors = FALSE)
str(trainData)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Tenim 891 observacions i 12 variables.

```
summary(trainData)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

Resum de les variables:

PassengerId (int): identificador del passatger

Survived (int): indica si el passatger va sobreviure (1) o no (0)

Pclass (int): classe en què viatjava el passatger (1, 2, 3)

Name (chr): nom

Sex (chr): male o female

Age (int): edat en anys

SibSp (int): número de fills i esposes a bord

Parch (int): número de pares i mares

Ticket (chr): número de ticket

Fare (int): preu del ticket

Cabin (chr): número de cabina

Embarked (chr): lloc d'embarcament (C, Q, S)

Notes sobre les dades:

edat: l'edat és fraccionada si és inferior a 1. Si s'estima l'edat, és en forma de xx.5

sibsp: El conjunt de dades defineix les relacions familiars d'aquesta manera ... Germà = germà, germana, germanastre, germanastra Cònjuge = marit, dona (les amants i els promès van ser ignorats)

parch: el conjunt de dades defineix les relacions familiars d'aquesta manera ... Parent = mare, pare Nen = filla, fill, fillastra, fillastre Alguns nens només viatjaven amb una mainadera, per tant, parch = 0 per a ells.

Exercici 3:

Neteja de les dades. Les dades contenen zeros o elements buits? Com gestionaries aquests casos?

```
# Registres amb valor NA
colSums(is.na(trainData))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

```
# Registres amb valor buit
colSums(trainData=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

Assignem valor "Desconeguda" per als valors buits de la variable "Cabin"

```
trainData$Cabin[trainData$Cabin==""] <- "Desconeguda"
head(trainData$Cabin,10)
```

```
##  [1] "Desconeguda" "C85"         "Desconeguda" "C123"        "Desconeguda"
##  [6] "Desconeguda" "E46"         "Desconeguda" "Desconeguda" "Desconeguda"
```

Assignem la mitjana per a valors buits de la variable "Age"

```
trainData$Age[is.na(trainData$Age)] <- signif(mean(trainData$Age,na.rm=T), digits=2)
head(trainData$Age,10)
```

```
##  [1] 22 38 26 35 35 30 54  2 27 14
```

Assignem NA als valors buits de Embarked:

```
trainData$Embarked[trainData$Embarked==""] <- NA
head(trainData$Embarked,20)
```

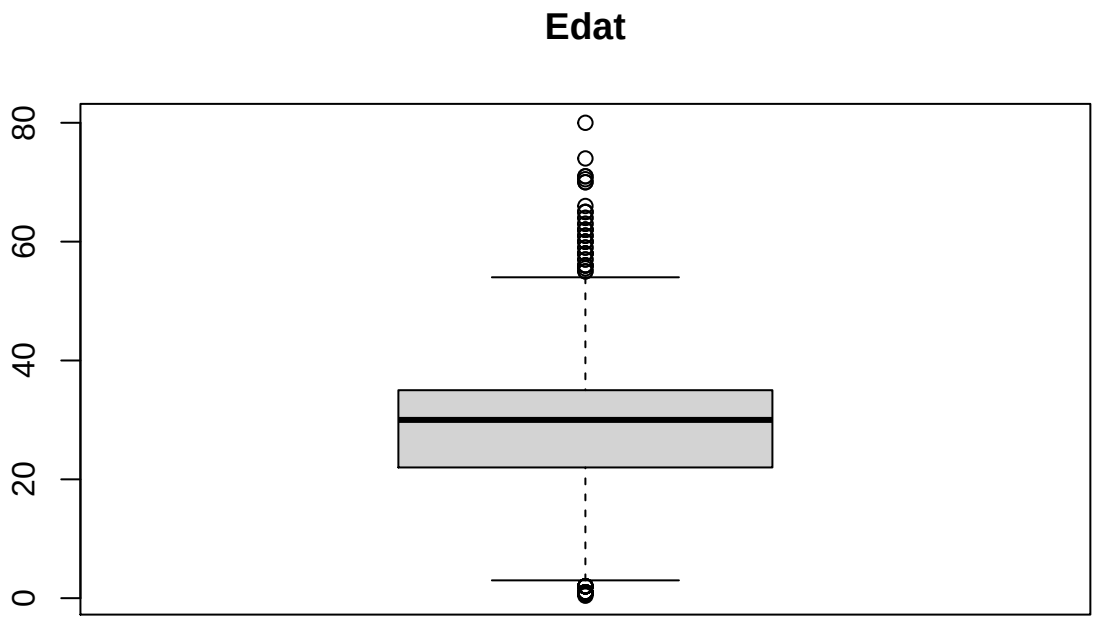
```
##  [1] "S" "C" "S" "S" "S" "Q" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S" "Q" "S" "S"
## [20] "C"
```

```
tail(trainData$Embarked,20)
```

```
##  [1] "S" "S" "S" "C" "C" "S" "S" "S" "C" "S" "S" "S" "S" "S" "Q" "S" "S" "S" "C"
## [20] "Q"
```

Identificació i tractament de valors extrems:

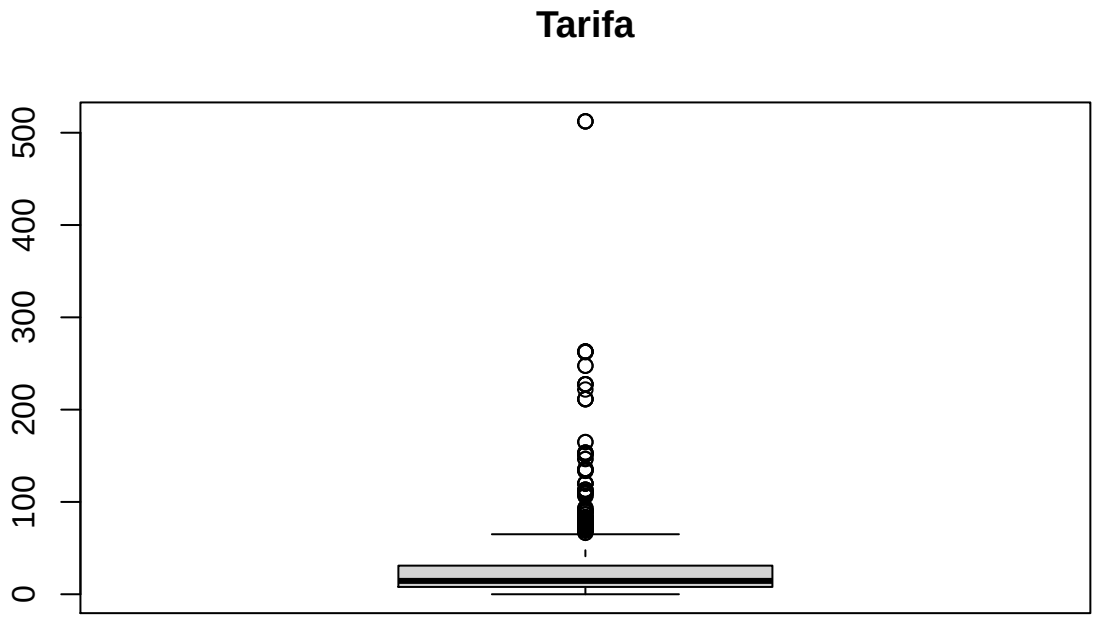
```
Age.bp<-boxplot(trainData$Age,main="Edat") # En Edat es representen 8 ouliers (66.0 71.0 70.5 71.0 80.0
```




```
head(Age.bp$out,8)
```

```
## [1]  2.00 58.00 55.00  2.00 66.00 65.00  0.83 59.00
```

```
Fare.bp<-boxplot(trainData$Fare,main="Tarifa") # En Fare en surten alguns més, però en hi ha un en conc
```



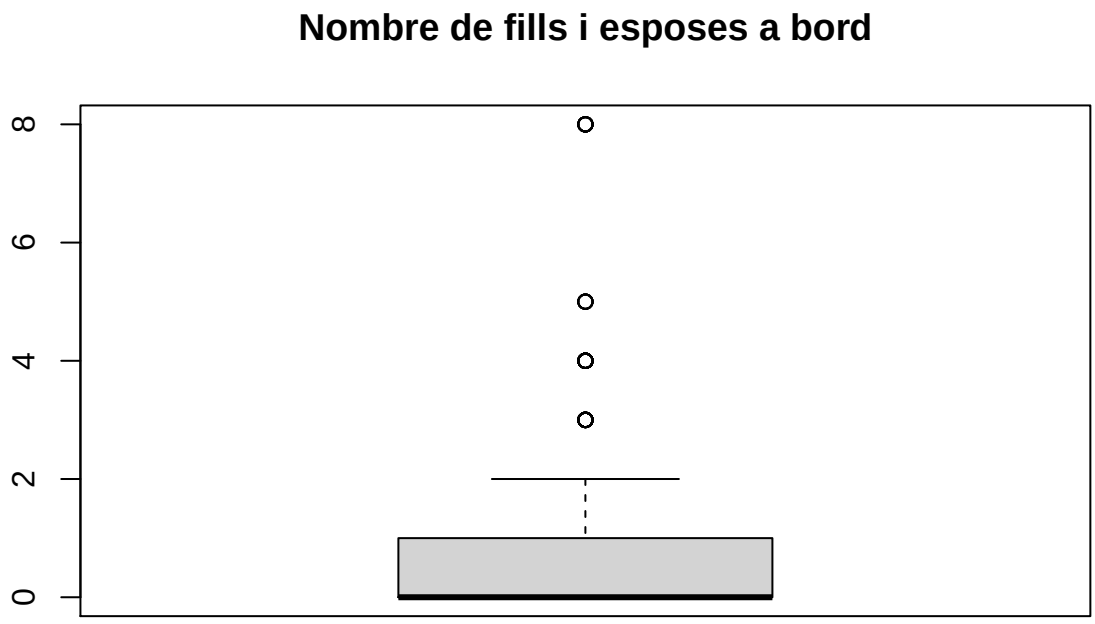

```
head(Fare.bp$out,10)
```

```
##  [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000
##  [9] 263.0000  77.2875
```

```
outlier_max<-max(Fare.bp$out,10)
outlier_max
```

```
## [1] 512.3292
```

```
SibSp.bp<-boxplot(trainData$SibSp,main="Nombre de fills i esposes a bord") # En aquesta variable hi ha
```




```
head(SibSp.bp$out,8)
```

```
## [1] 3 4 3 3 4 5 3 4
```

```
# En aquest cas, no hi hauria que tractar els valors extrems, ja que no distorsionen els resultats de l
```

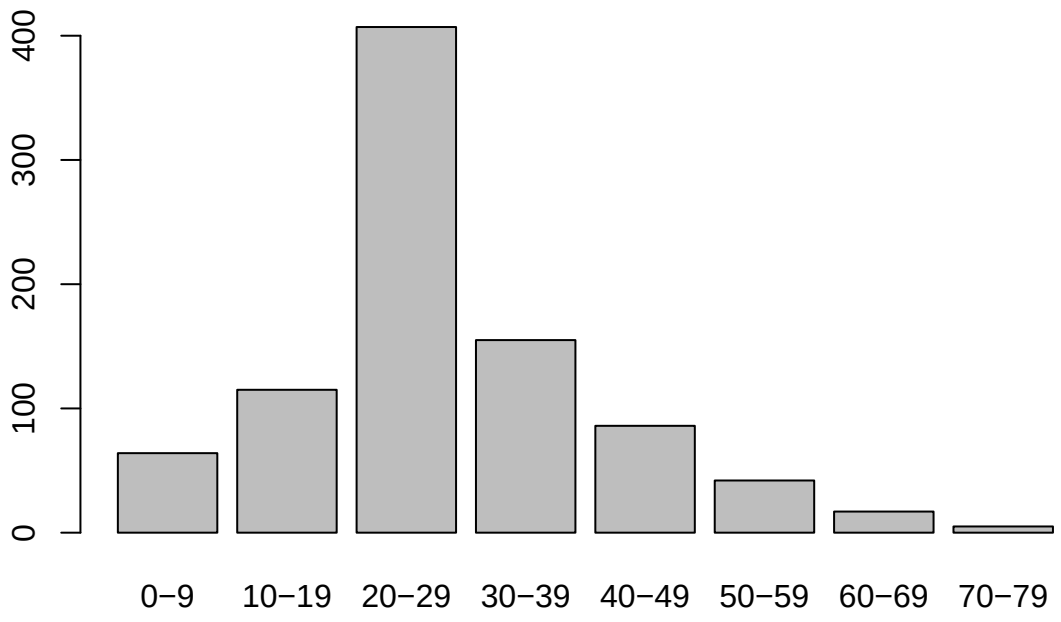
El outlier amb valor màxim de la variable Fare és 512.3292.

Discretització variable edat en intervals:

```
trainData["segment_edat"] <- cut(trainData$Age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c("0-9
```

Gràfic:

```r
plot(trainData$segment_edat)
```



**** # Exercici 4: **** **** ## Mètode d'agregació: ****

```r
#Llibreria cluster per fer agrupacions
library(cluster)
```

La funció daisy() que utilitzarem per calcular la silueta de la mostra només funciona amb valors numèrics i l'atribut Sex és un string. Per solucionar aquest inconvenient farem un one-hot encoding transformant el Sex en dos nous atributs binaris:

```r
library(caret)
dummies <- predict(dummyVars(~ Sex, data = trainData), newdata = trainData)
trainData <- cbind(trainData, dummies)
summary(trainData)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
```

```
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:22.00   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :30.00   Median :0.000   Median :0.0000
##                     Mean   :29.76   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:35.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
##   segment_edat   Sexfemale         Sexmale
##  20-29  :407   Min.   :0.0000   Min.   :0.0000
##  30-39  :155   1st Qu.:0.0000   1st Qu.:0.0000
##  10-19  :115   Median :0.0000   Median :1.0000
##  40-49  : 86   Mean   :0.3524   Mean   :0.6476
##  0-9    : 64   3rd Qu.:1.0000   3rd Qu.:1.0000
##  50-59  : 42   Max.   :1.0000   Max.   :1.0000
##  (Other): 22
```

Els camps que utilitzarem per fer les agrupacions són: Survived, Pclass, Sexmale, Sexfemale i segment_edat:

```
clustering_data <- trainData[ , c("Survived", "Pclass", "Sexfemale", "Sexmale", "Age")]
str(clustering_data)
```

```
## 'data.frame':    891 obs. of  5 variables:
##  $ Survived : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass   : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Sexfemale: num  0 1 1 1 0 0 0 0 1 1 ...
##  $ Sexmale  : num  1 0 0 0 1 1 1 1 0 0 ...
##  $ Age      : num  22 38 26 35 35 30 54 2 27 14 ...
```

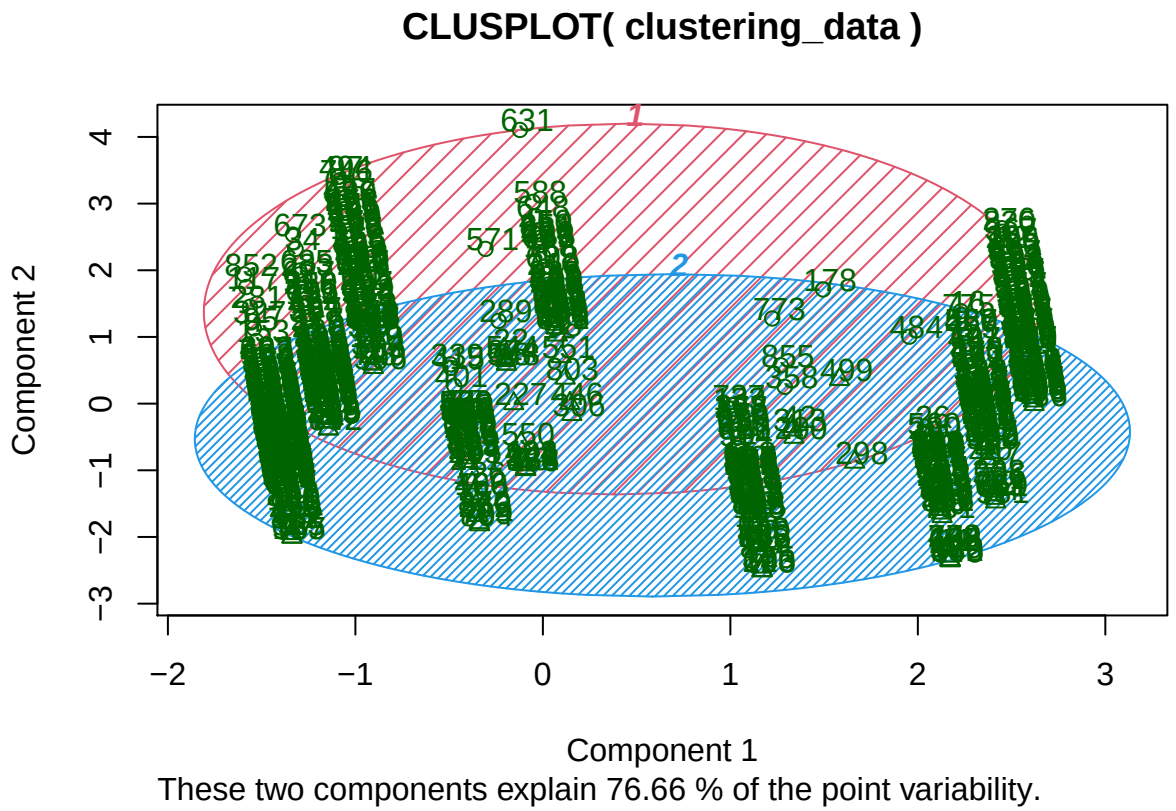
Passem a executar l'algorisme kmeans, com que inicialment no coneixem el nombre de clusters, provem d'aplicar l'algorisme amb 2, 3, 4, 5, 6, 7 i 8 clústers.

```
clustering_data2       <- kmeans(clustering_data, 2)
passatgers_cluster2 <- clustering_data2$cluster

clustering_data3       <- kmeans(clustering_data, 3)
passatgers_cluster3 <- clustering_data3$cluster

clustering_data4       <- kmeans(clustering_data, 4)
passatgers_cluster4 <- clustering_data4$cluster

clustering_data5       <- kmeans(clustering_data, 5)
passatgers_cluster5 <- clustering_data5$cluster

clustering_data6       <- kmeans(clustering_data, 6)
passatgers_cluster6 <- clustering_data6$cluster
```

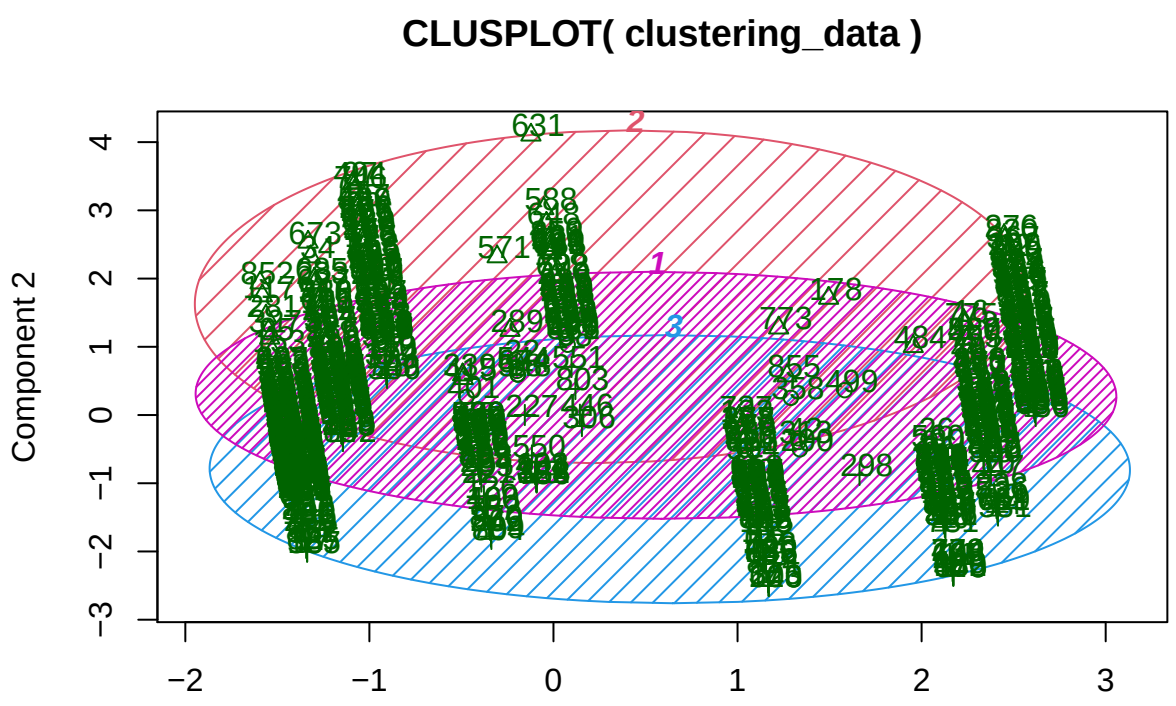
```r
clustering_data7       <- kmeans(clustering_data, 7)
passatgers_cluster7 <- clustering_data7$cluster

clustering_data8       <- kmeans(clustering_data, 8)
passatgers_cluster8 <- clustering_data8$cluster
```

Podem veure gràficament els clusters obtinguts amb la següent funció:

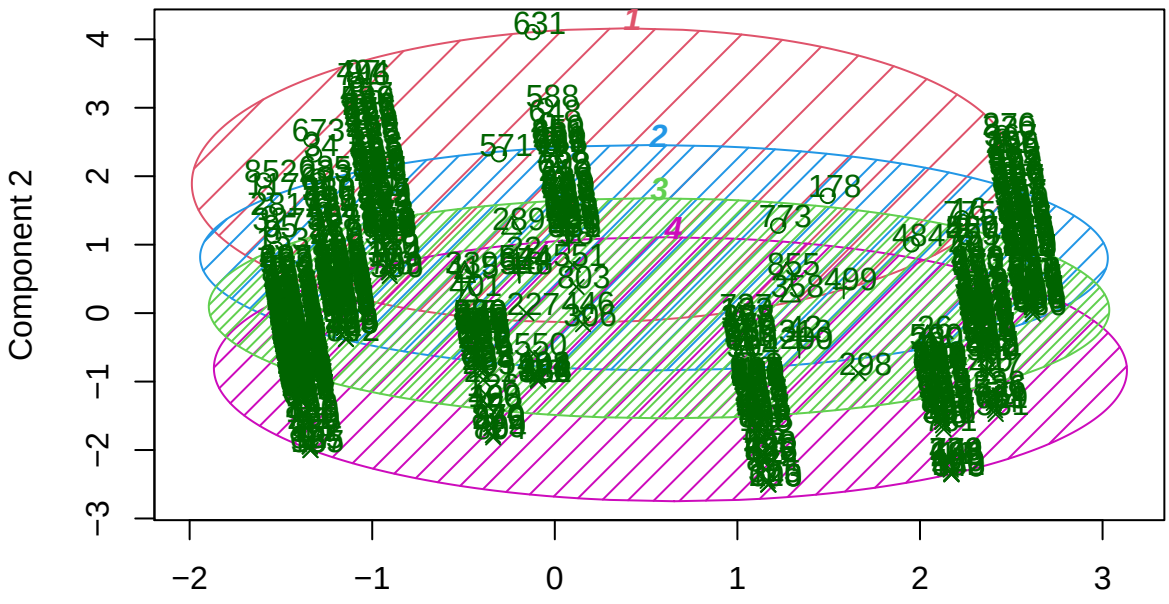
```r
clusplot(clustering_data, clustering_data2$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



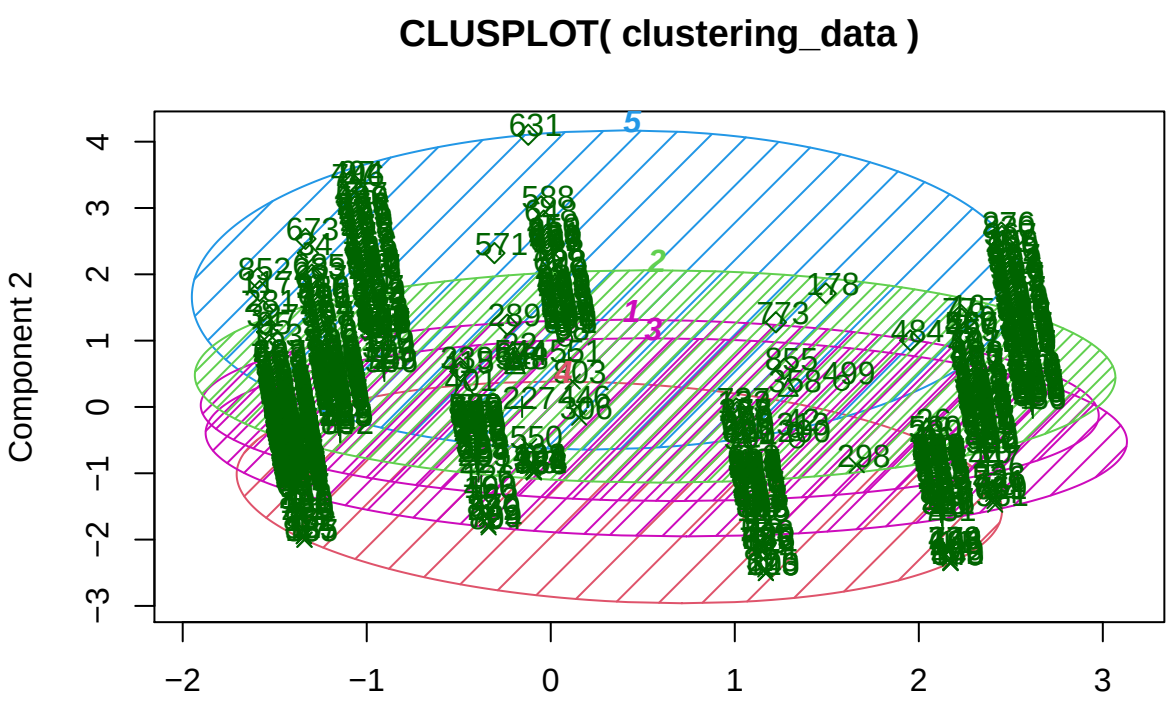

```r
clusplot(clustering_data, clustering_data3$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



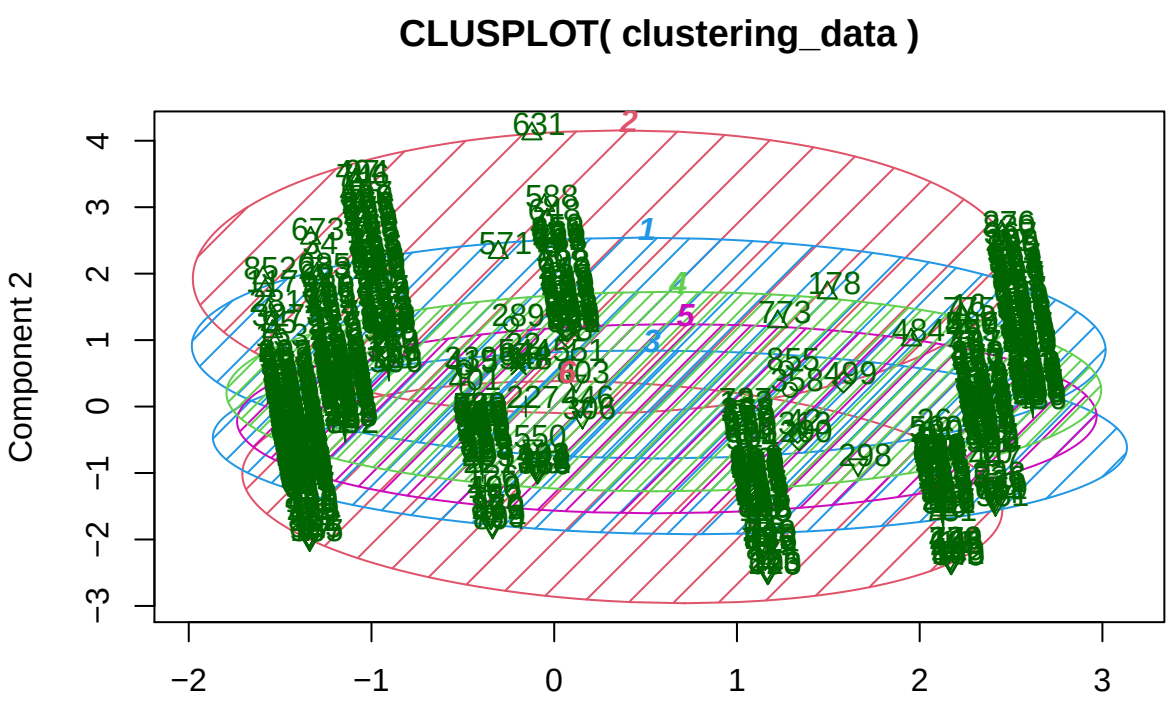

```
clusplot(clustering_data, clustering_data4$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



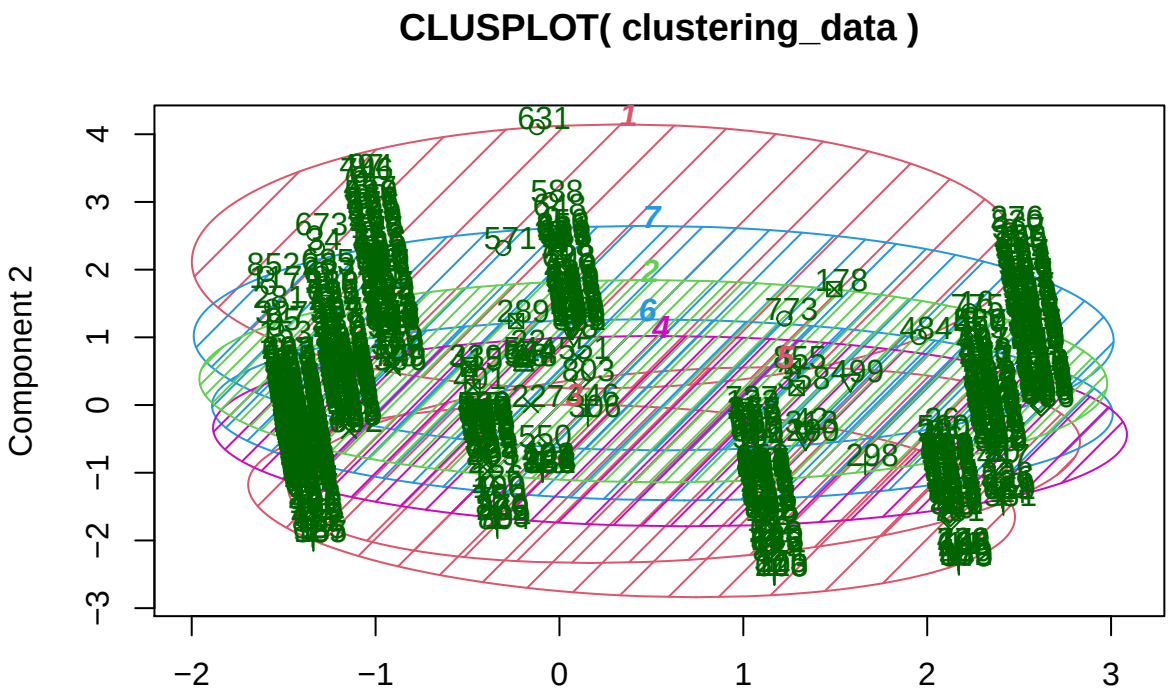

```
clusplot(clustering_data, clustering_data5$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



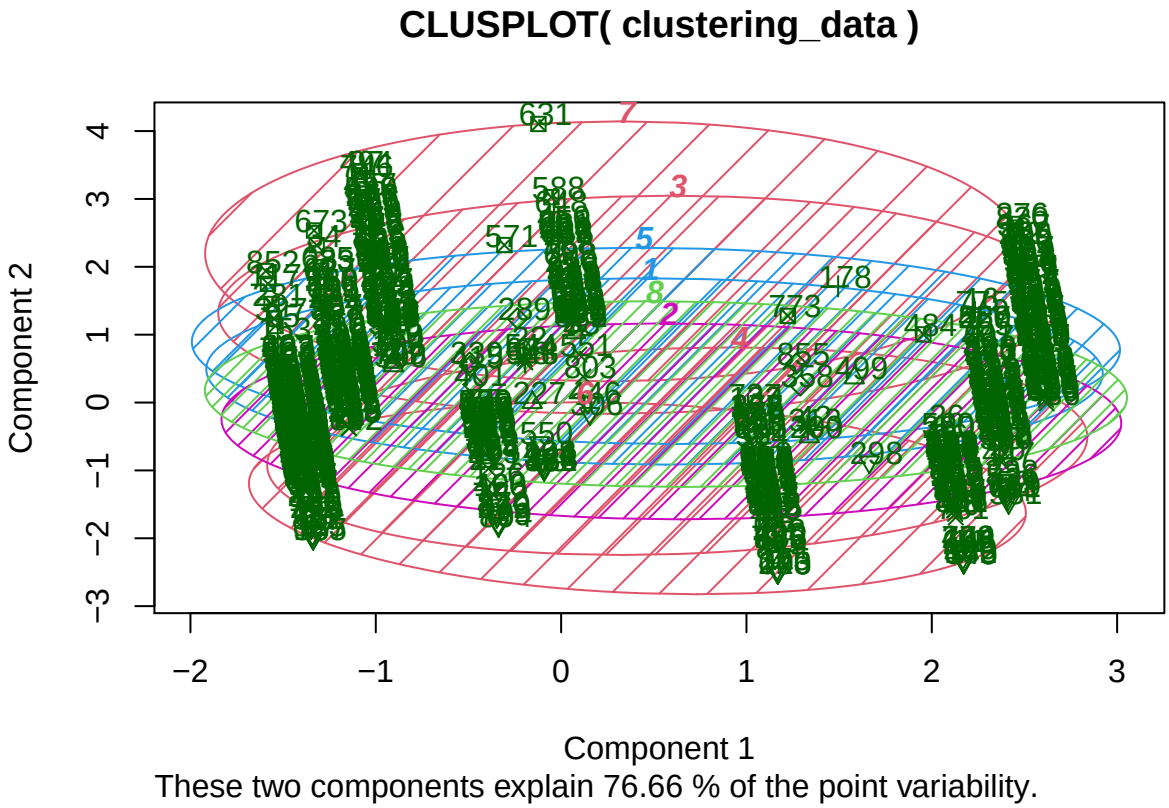

```
clusplot(clustering_data, clustering_data6$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```




```
clusplot(clustering_data, clustering_data7$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```




```
clusplot(clustering_data, clustering_data8$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```

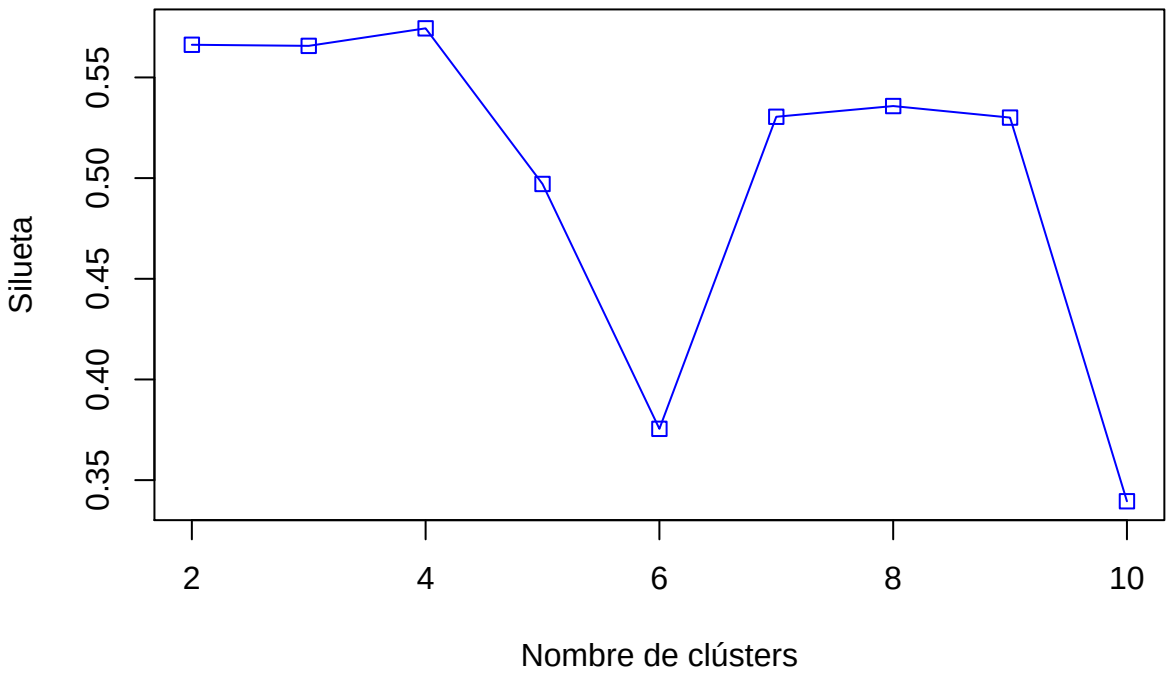


Ara calcularem la silueta de les mostres per avaluar la qualitat del mètode d'agregació.

```
d <- daisy(clustering_data)
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(clustering_data, i)
  y_cluster     <- fit$cluster
  sk            <- silhouette(y_cluster, d)
  resultados[i] <- mean(sk[,3])
}
```

Mostrem en un gràfica els valors de les siluetes mitjana de cada prova per a comprovar quin nombre de clústers és el millor.

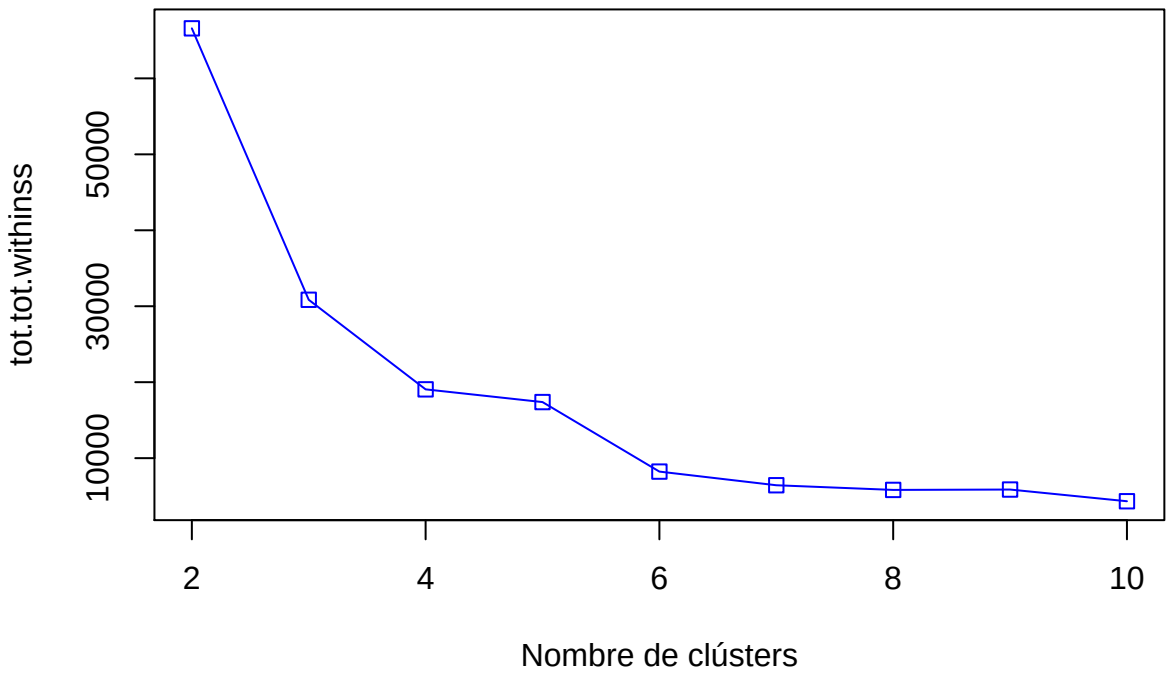
```
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Nombre de clústers",ylab="Silueta")
```



Veiem que la millor agrupació és amb 5 clusters i la segona millor amb 4.

Per comparar resultats, provem de fer l'avaluació del millor nombre de clusters amb la funció withinss.

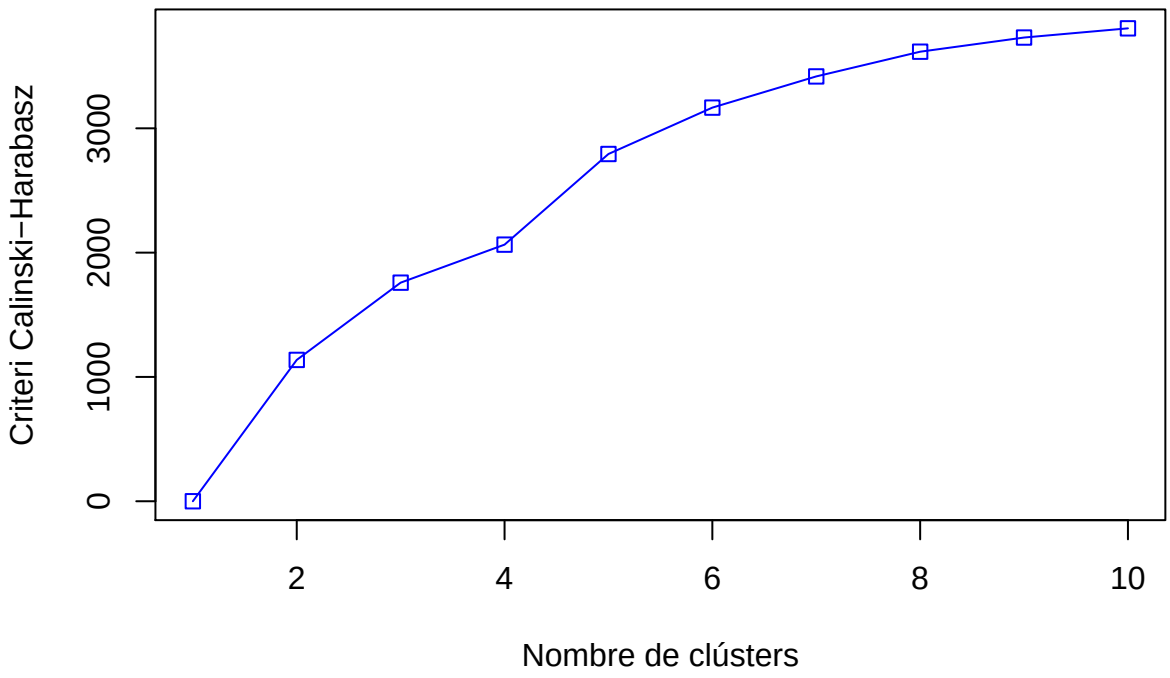
```
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(clustering_data, i)
  resultados[i] <- fit$tot.withinss
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Nombre de clústers",ylab="tot.tot.withinss")
```



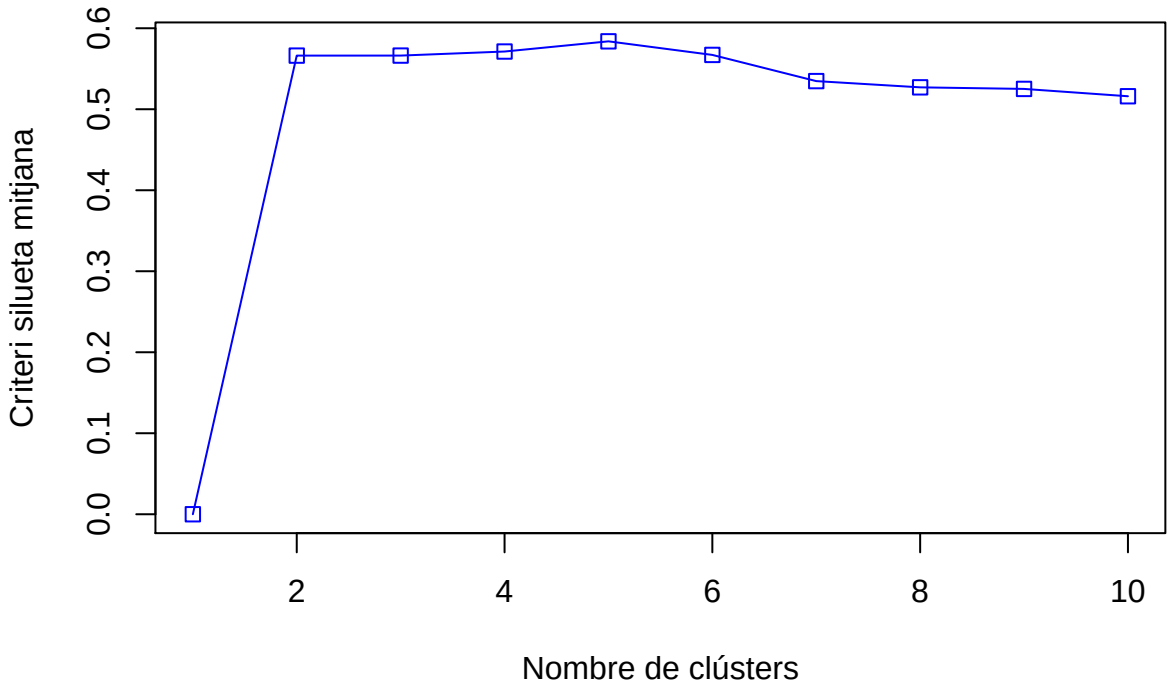
En aquest cas és difícil trobar el millor valor de k perquè el gràfic té una corba molt rodona i no hi ha cap colze clar.

Per últim provarem de fer l'avaluació amb la funció kmeansruns utilitzant els criteris de silueta mitjana i de Calinski-Harabasz:

```
library(fpc)
fit_ch  <- kmeansruns(clustering_data, krange = 1:10, criterion = "ch")
fit_asw <- kmeansruns(clustering_data, krange = 1:10, criterion = "asw")

plot(1:10,fit_ch$crit,type="o",col="blue",pch=0,xlab="Nombre de clústers",ylab="Criteri Calinski-Haraba
```




```
plot(1:10,fit_asw$crit,type="o",col="blue",pch=0,xlab="Nombre de clústers",ylab="Criteri silueta mitjana
```



També sembla que el punt més alt és per k=5, com en el primer cas.

Mètode d'associació

Arbres de decisió

Per la visualització gràfica de les variables utilitzarem els paquets ggplot2, gridExtra i grid de R.

```
if(!require(ggplot2)){
    install.packages('ggplot2', repos='http://cran.us.r-project.org')
    library(ggplot2)
}
if(!require(ggpubr)){
    install.packages('ggpubr', repos='http://cran.us.r-project.org')
    library(ggpubr)
}
if(!require(grid)){
    install.packages('grid', repos='http://cran.us.r-project.org')
    library(grid)
}
if(!require(gridExtra)){
    install.packages('gridExtra', repos='http://cran.us.r-project.org')
    library(gridExtra)
}
if(!require(C50)){
    install.packages('C50', repos='http://cran.us.r-project.org')
```

```
    library(C50)
}
```

A continuació construim l'arbre de decisió a partir del dataset d'entrenament. A la funció li passem com a primer paràmetre el subconjunt d'entrenament excloent el camp 'Survived' (train[-2]) i com a segon paràmetre el propi camp (trainData$Survived):

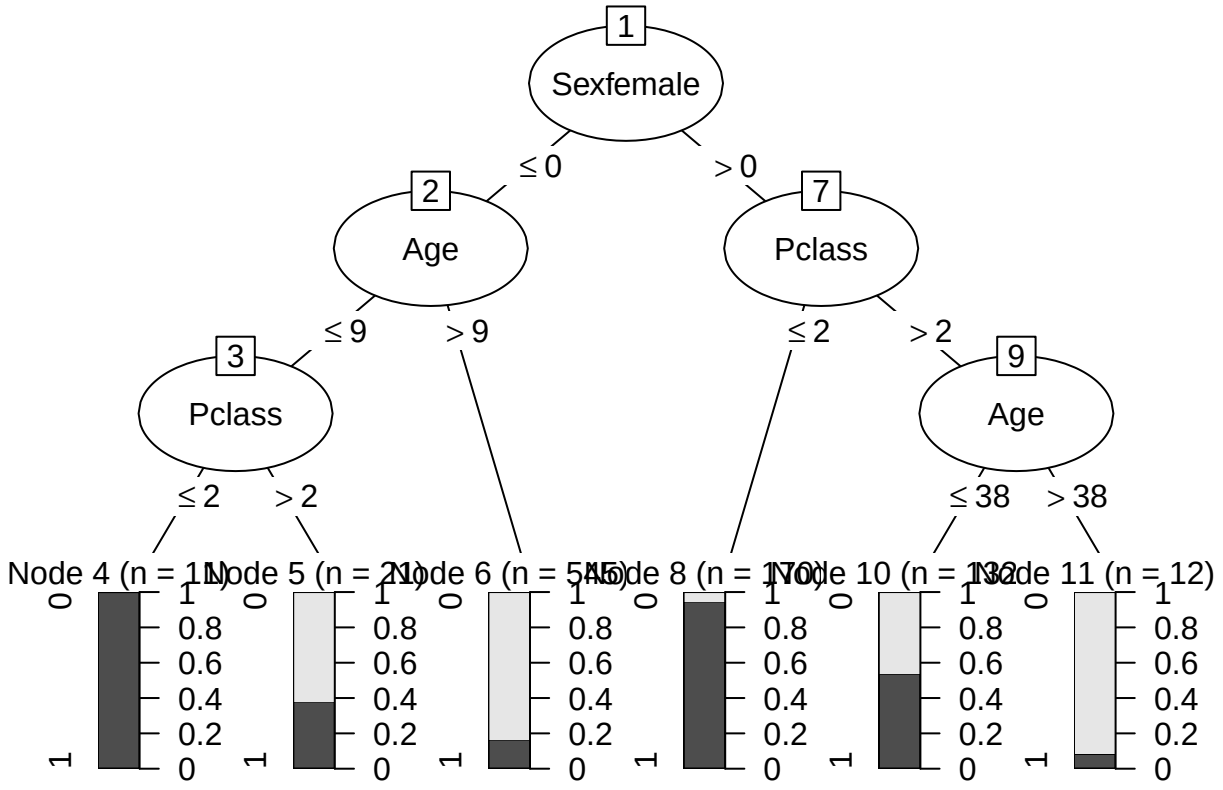
```
set.seed(891)
clustering_data$Survived = as.factor(clustering_data$Survived)
model <- C50::C5.0(clustering_data[-1], clustering_data$Survived)
summary(model)
```

```
##
## Call:
## C5.0.default(x = clustering_data[-1], y = clustering_data$Survived)
##
##
## C5.0 [Release 2.07 GPL Edition]      Fri May 21 10:45:07 2021
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 891 cases (5 attributes) from undefined.data
##
## Decision tree:
##
## Sexfemale <= 0:
## :...Age > 9: 0 (545/90)
## :   Age <= 9:
## :   :...Pclass <= 2: 1 (11)
## :       Pclass > 2: 0 (21/8)
## Sexfemale > 0:
## :...Pclass <= 2: 1 (170/9)
##     Pclass > 2:
##     :...Age <= 38: 1 (132/61)
##         Age > 38: 0 (12/1)
##
##
## Evaluation on training data (891 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       6  169(19.0%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     479    70    (a): class 0
##      99   243    (b): class 1
##
##
```

```
##  Attribute usage:
##
##  100.00% Sexfemale
##   80.92% Age
##   38.83% Pclass
##
##
## Time: 0.0 secs
```

La visualització de l'arbre obtingut no es pot interpretar perquè hi ha masses camps:

```
plot(model)
```



Ara descomposarem l'arbre en un set de regles amb el flag rules=TRUE:

```
model2 <- C50::C5.0(clustering_data[-1], clustering_data$Survived, rules = TRUE)
summary(model2)
```

```
##
## Call:
## C5.0.default(x = clustering_data[-1], y = clustering_data$Survived, rules
##  = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Fri May 21 10:45:07 2021
## -----------------------------
```

```
##
## Class specified by attribute 'outcome'
##
## Read 891 cases (5 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (51/4, lift 1.5)
##  Pclass > 2
##  Age > 38
##  ->  class 0  [0.906]
##
## Rule 2: (577/109, lift 1.3)
##  Sexfemale <= 0
##  ->  class 0  [0.810]
##
## Rule 3: (11, lift 2.4)
##  Pclass <= 2
##  Sexfemale <= 0
##  Age <= 9
##  ->  class 1  [0.923]
##
## Rule 4: (314/81, lift 1.9)
##  Sexfemale > 0
##  ->  class 1  [0.741]
##
## Default class: 0
##
##
## Evaluation on training data (891 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       4  169(19.0%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     479    70    (a): class 0
##      99   243    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% Sexfemale
##    6.96% Pclass
##    6.96% Age
##
##
## Time: 0.0 secs
```

Explicació de les regles:

....

Ara passem a avaluar la qualitat del primer model creat. Carreguem el dataset de test:

```
testData <- read.csv('data/test.csv',stringsAsFactors = FALSE)
#test_data <- testData[ , c("Survived", "Pclass", "Sexfemale", "Sexmale", "Age")]
```

COM AVALUAR????

```
#predicted_model <- predict( model, test[-17], type="class" )
#print(sprintf("La precisió de l'arbre és del %.4f %%",100*sum(predicted_model == test$default) / lengt
```

Exercici 5:

Exercici 6:

Contribucions a la pràctica

```
tab <- matrix(c('Vicenç i Begoña', 'Vicenç i Begoña', 'Vicenç i Begoña'), ncol=1, byrow=TRUE)
colnames(tab) <- c('Firma')
rownames(tab) <- c('Investigació prèvia','Redacció de les respostes','Desenvolupament codi')
tab <- as.table(tab)
tab
```

```
##                           Firma
## Investigació prèvia       Vicenç i Begoña
## Redacció de les respostes Vicenç i Begoña
## Desenvolupament codi      Vicenç i Begoña
```