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.

Disponsem 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 disponsem. 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.

Aquests dos conjunts de dades estan creats aleatòriament a partir de les dades oficials de passatgers 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?

Comprovem el nombre d'elements buits del joc de dades:

```
# 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
```

Veiem que els camps edat, cabina i embarcat contenen valors buits. Aquesta inexactitud de les dades és degut probablement a que el registre de passatgers no va ser del tot rigurós en aquell moment. Alguns dels passatgers es van colar a bord sense tenir el bitllet, això fa que no es tinguessin dades de l'edat ni de la cabina que tenien.

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

Enlloc de un string en blanc, assignem la paraula "Desconeguda"

```
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":

En el cas de l'edat assignem la mitjana per no alterar les dades.

```
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:

I pel cas dels camps buits d'Embarked, assignem NA.

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

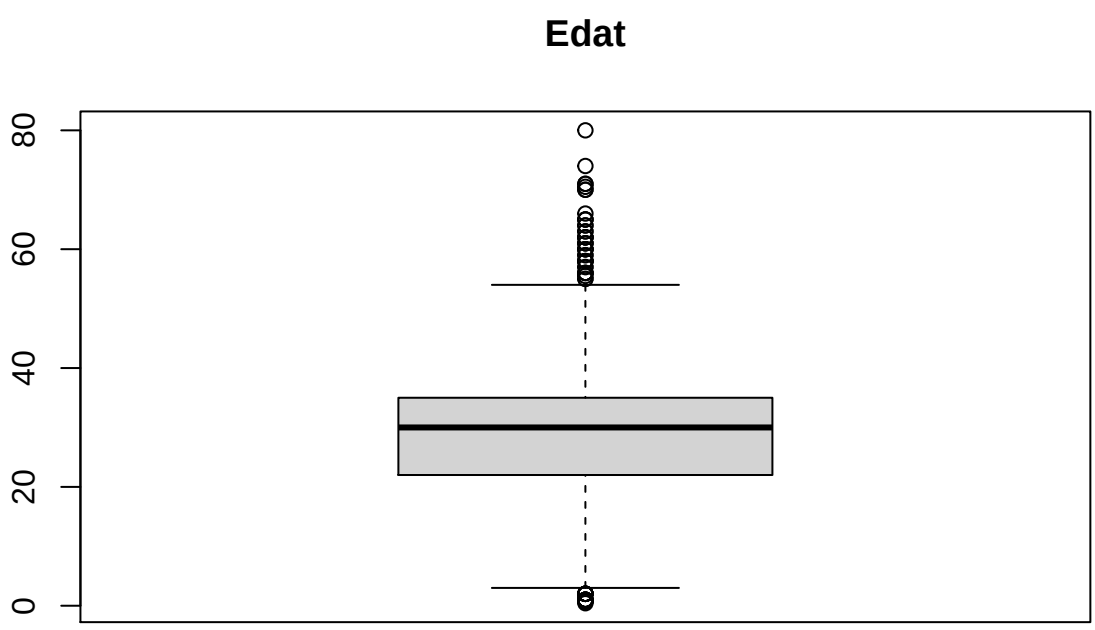
```
##  [1] "S" "C" "S" "S" "S" "Q" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S" "Q" "S" "S"
## [20] "C"
```

```
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:

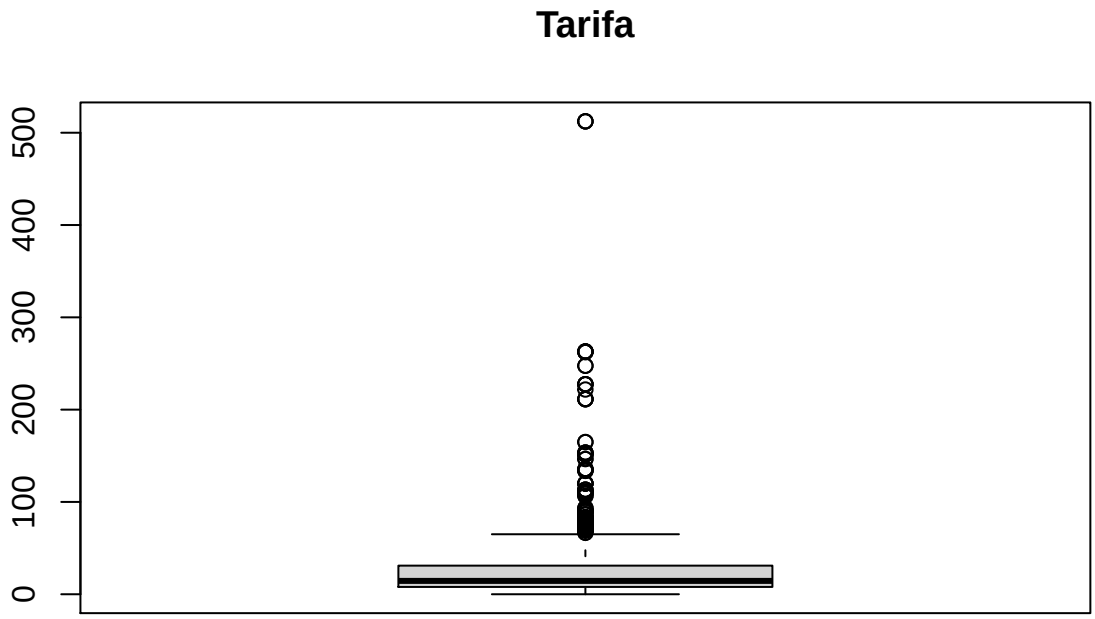
```
Age.bp<-boxplot(trainData$Age,main="Edat")
```




```
#En la variable Edat es representen 8 outliers (66.0 71.0 70.5 71.0 80.0 70.0 70.0 74.0). Aquest valors
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")
```



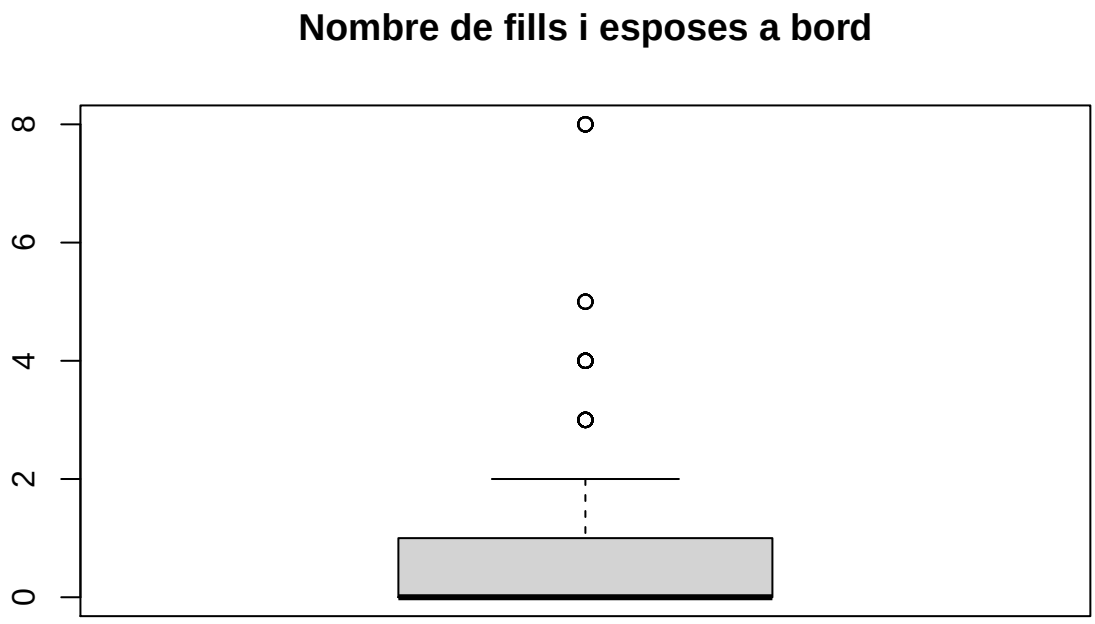

```
#En la variable Fare en surten alguns més, però n'hi ha un en concret molt lluny de la resta.
head(Fare.bp$out,10)
```

```
##  [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 4 outliers,que no vol dir res més que se surten de la mitjana dels valors de
head(SibSp.bp$out,8)
```

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

En aquest cas, no caldria tractar els valors extrems ja que no distorsionen els resultats de les prediccions que volem fer amb la base de dades. Tot i ser valors que surten de la mitjana, no són incorrectes ni errades. L'outlier amb valor màxim de la variable Fare és 512.3292.

Exercici 4:

Mètode d'agregació:

En aquest apartat farem un anàlisi de les dades utilitzant un mètode d'agregació. Obtindrem grups (clusters) que agrupen les dades segons la semblança entre elles. Primer de tot importem la llibreria:

```
#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:

```
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
##    Sexfemale         Sexmale
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :1.0000
##  Mean   :0.3524   Mean   :0.6476
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

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

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

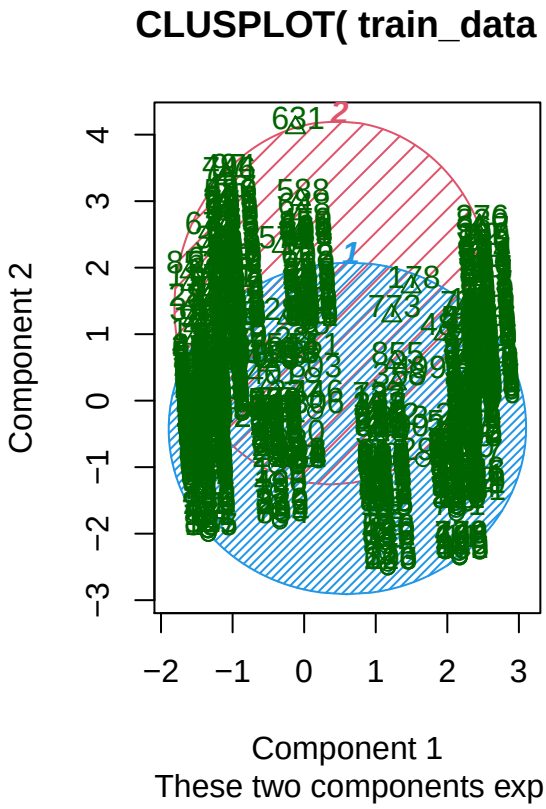
```
##  $ Sexfemale: num  0 1 1 1 0 0 0 0 1 1 ...
##  $ Sexmale  : num  1 0 0 0 1 1 1 1 0 0 ...
##  $ Age      : num  22 38 26 35 35 30 54 2 27 14 ...
```

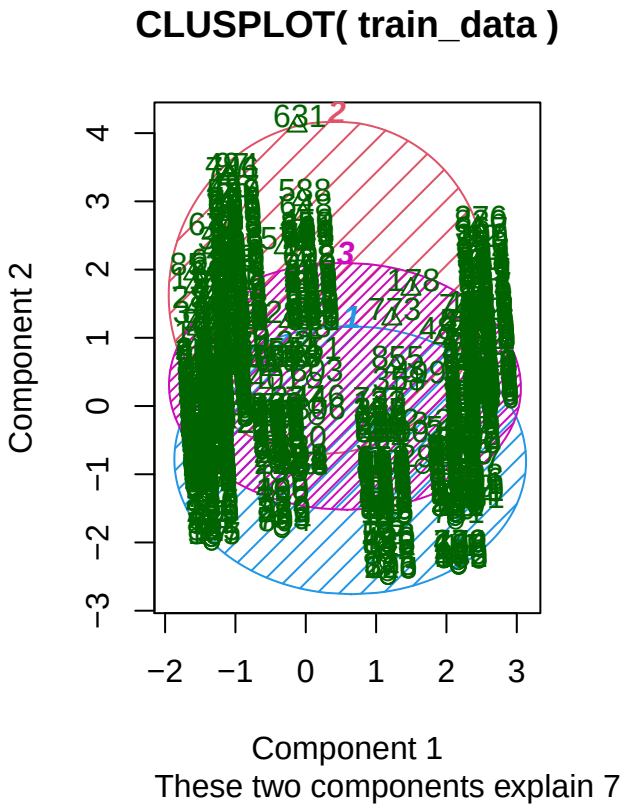
Passem a executar l'algorisme kmeans, com que inicialment no coneixem el nombre de clusters, provem d'aplicar l'algorisme amb 2, 3, 4, 5, 6, 7 i 8 clústers.

```
train_data2       <- kmeans(train_data, 2)
passatgers_cluster2 <- train_data2$cluster

train_data3       <- kmeans(train_data, 3)
passatgers_cluster3 <- train_data3$cluster

train_data4       <- kmeans(train_data, 4)
passatgers_cluster4 <- train_data4$cluster

train_data5       <- kmeans(train_data, 5)
passatgers_cluster5 <- train_data5$cluster

train_data6       <- kmeans(train_data, 6)
passatgers_cluster6 <- train_data6$cluster

train_data7       <- kmeans(train_data, 7)
passatgers_cluster7 <- train_data7$cluster

train_data8       <- kmeans(train_data, 8)
passatgers_cluster8 <- train_data8$cluster
```

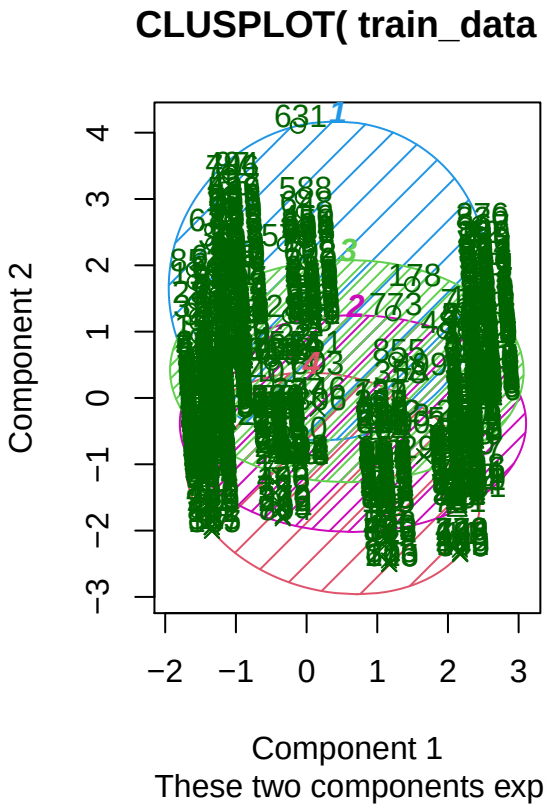
Podem veure gràficament els clusters obtinguts amb la següent funció:

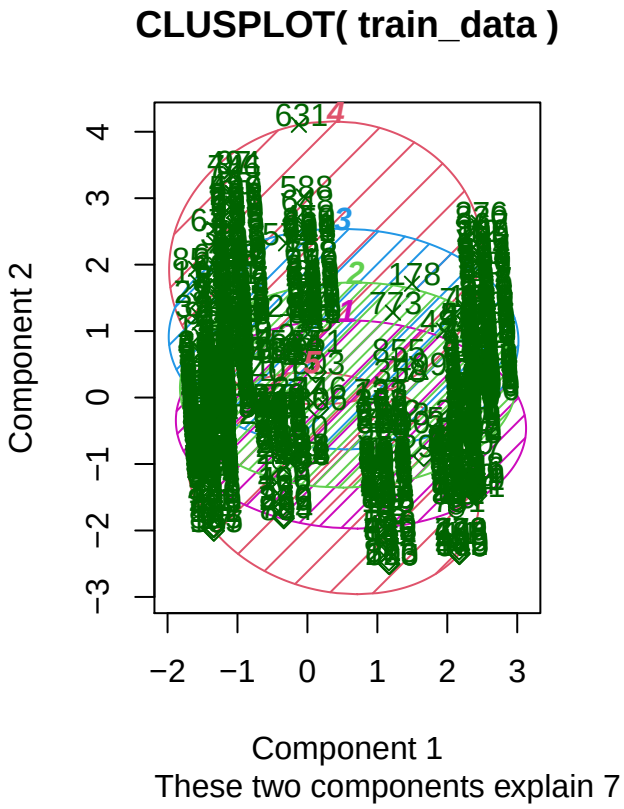
```
library(plotfunctions)
par(mfrow=c(1,2))

clusplot(train_data, train_data2$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
clusplot(train_data, train_data3$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



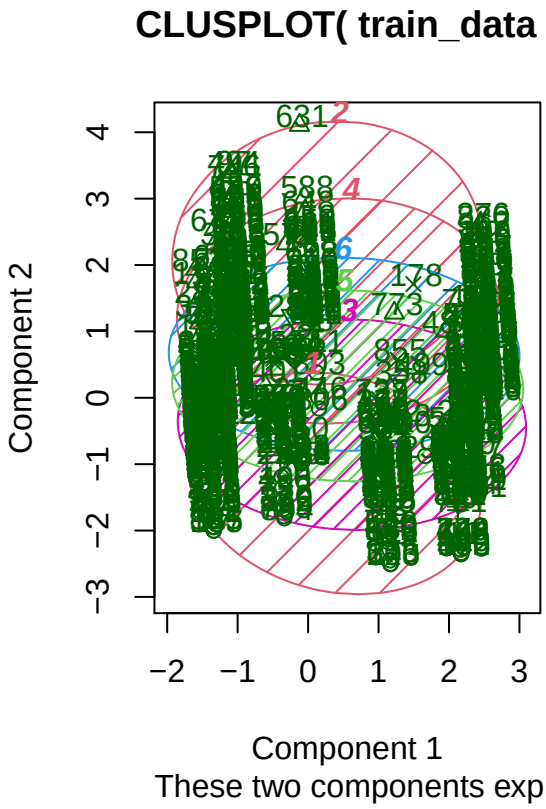


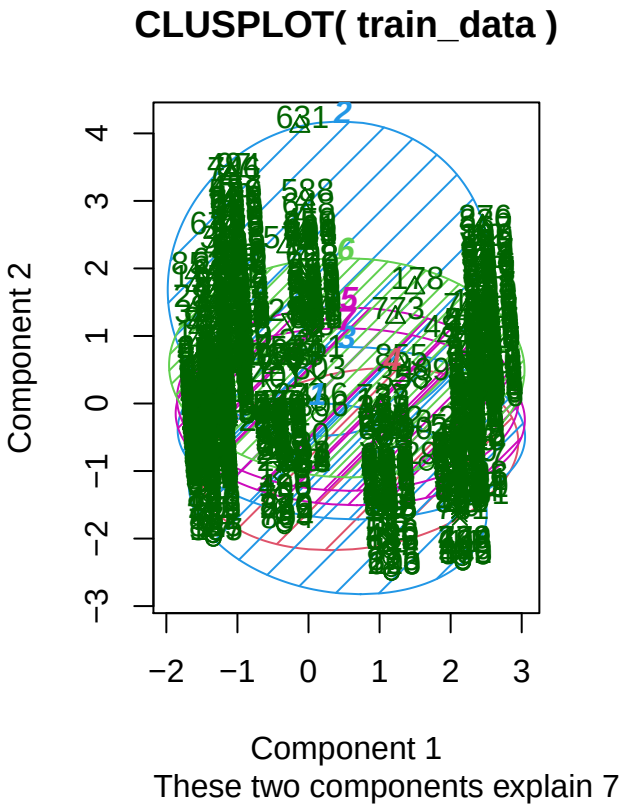

```
clusplot(train_data, train_data4$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
clusplot(train_data, train_data5$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



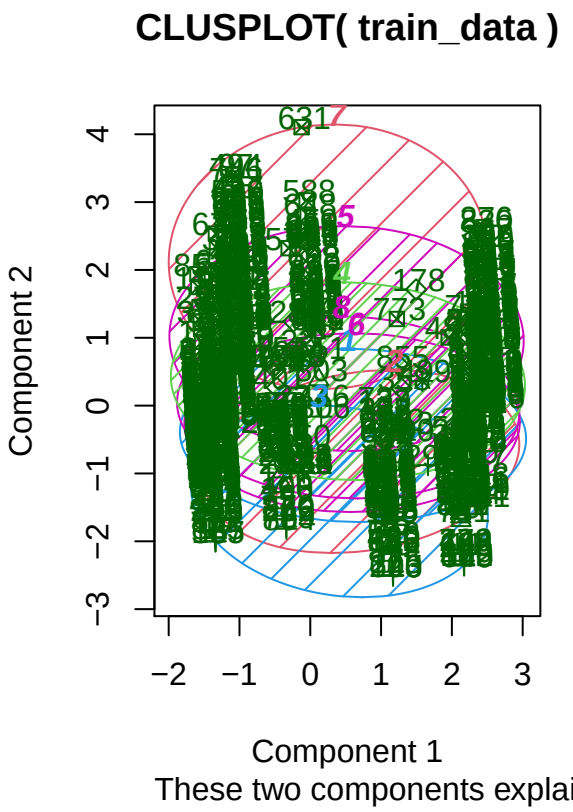



```
clusplot(train_data, train_data6$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
clusplot(train_data, train_data7$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```






```
clusplot(train_data, train_data8$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```

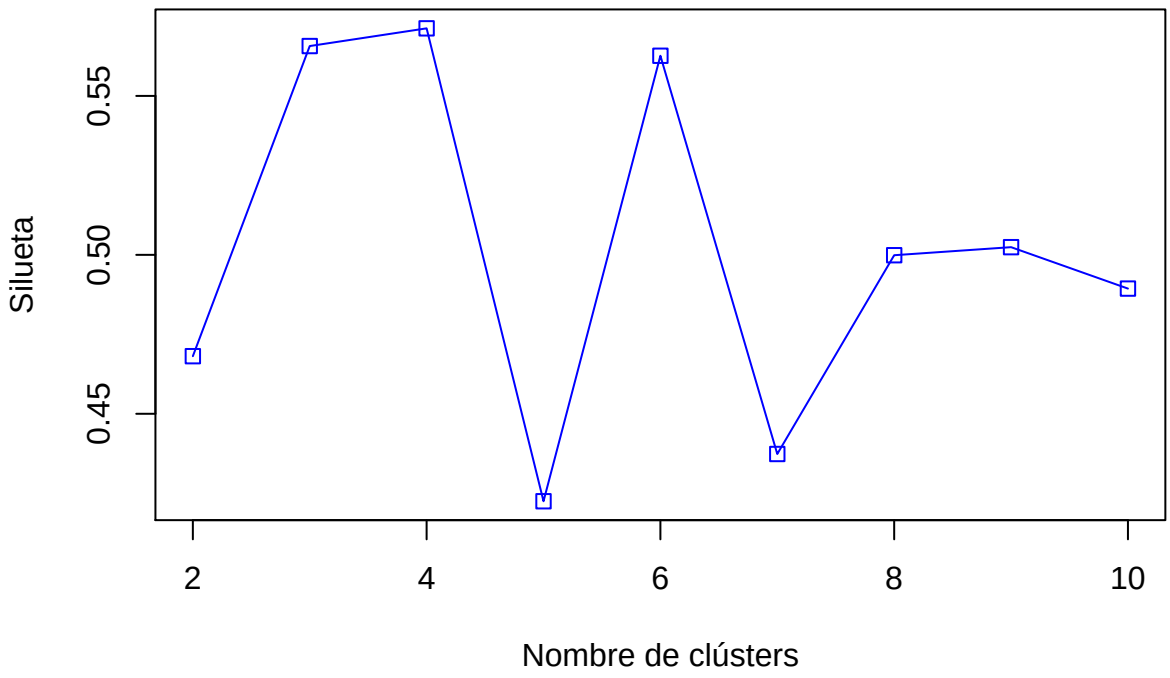


Ara calcularem la silueta de les mostres per avaluar la qualitat del mètode d'agregació.

```
set.seed(891)
d <- daisy(train_data)
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(train_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.

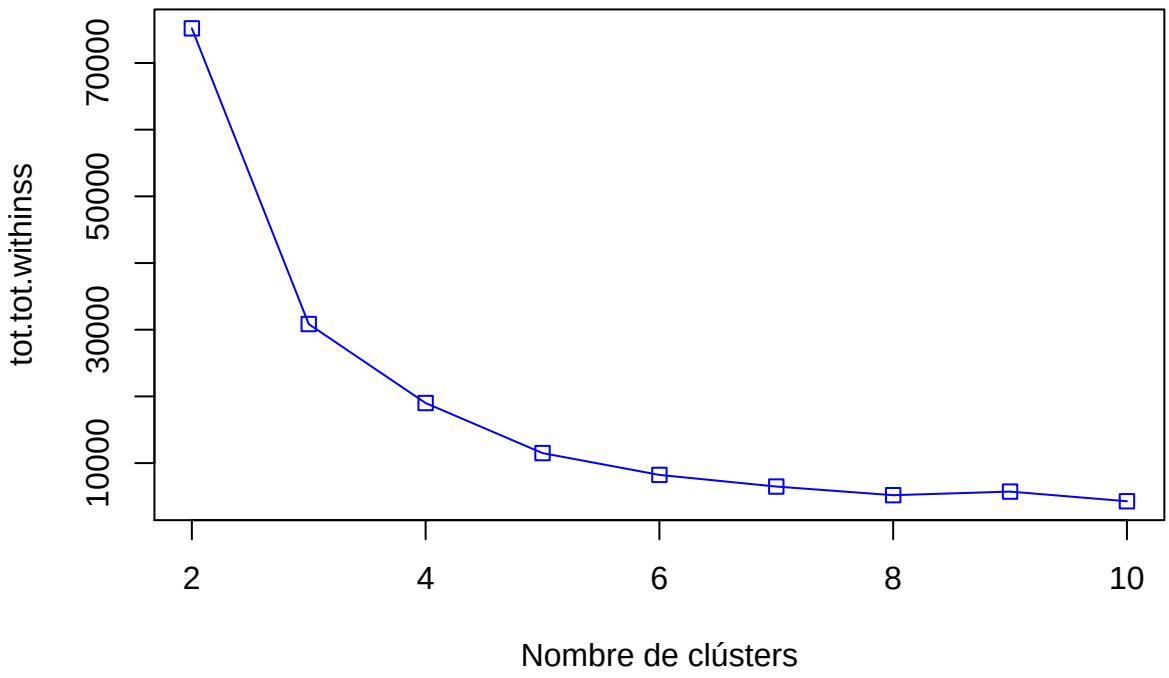
```
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Nombre de clústers",ylab="Silueta")
```



Veiem que la millor agrupació és amb 4 clusters i la segona millor amb 3.

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

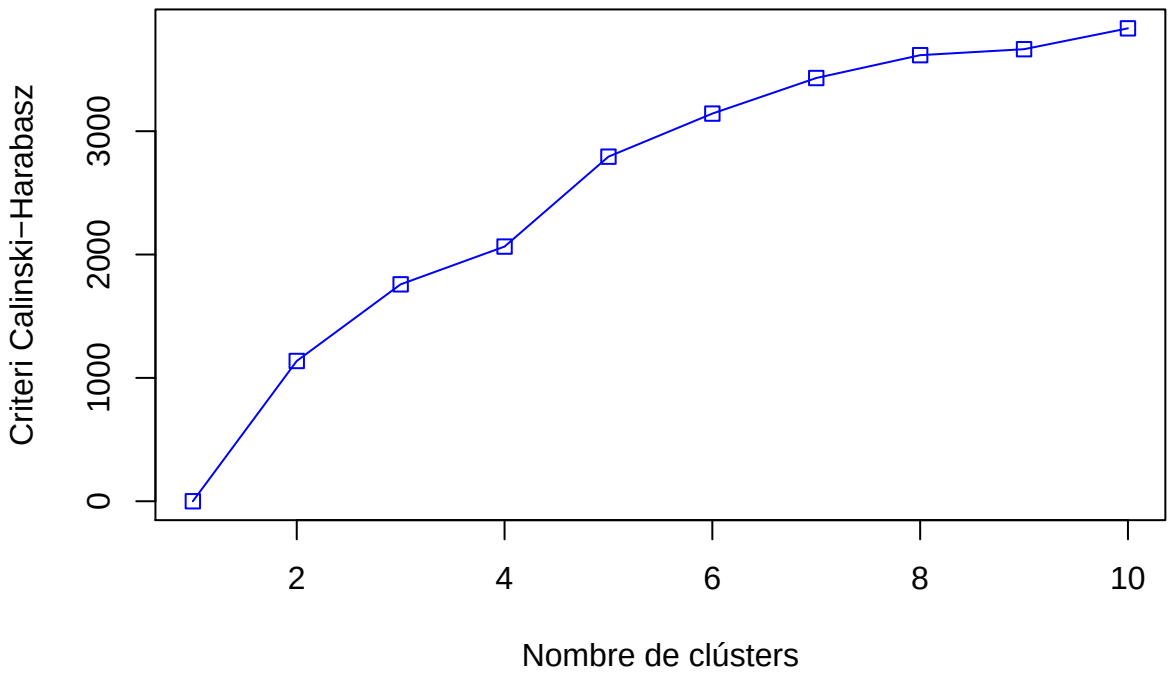
```
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(train_data, i)
  resultados[i] <- fit$tot.withinss
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Nombre de clústers",ylab="tot.tot.withinss")
```



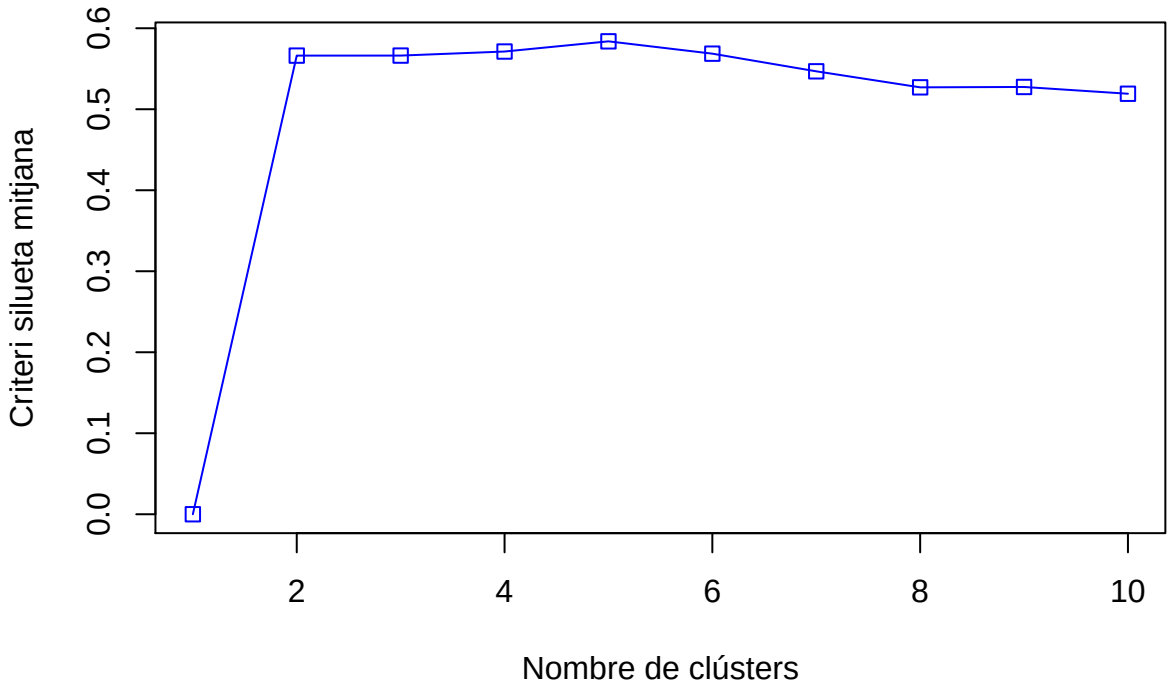
En aquesta funció hem de buscar el "colze" de la corba per tenir el millor valor de k. En aquest cas és difícil trobar el millor valor perquè el gràfic té una corba molt rodona i no hi ha cap colze clar, tot i que potser seria el 3 o el 4.

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(train_data, krange = 1:10, criterion = "ch")
fit_asw <- kmeansruns(train_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
```

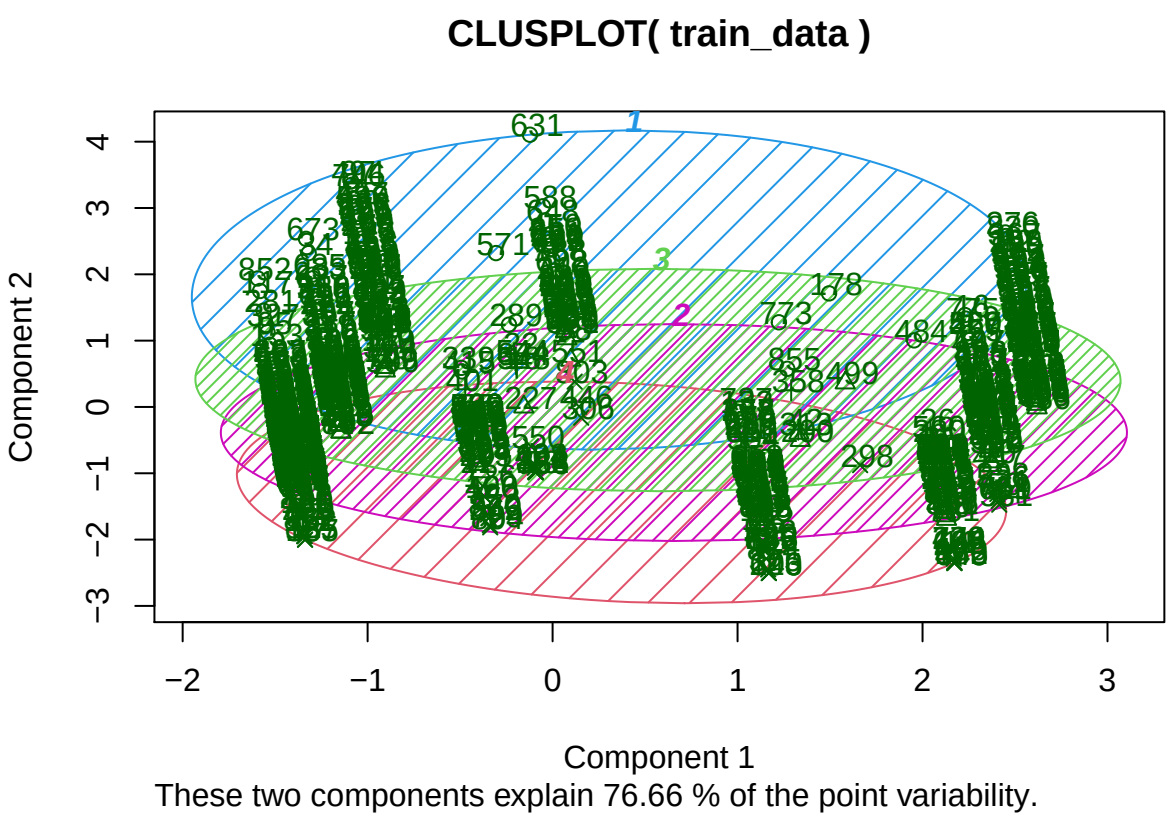



```
plot(1:10,fit_asw$crit,type="o",col="blue",pch=0,xlab="Nombre de clústers",ylab="Criteri silueta mitjana
```



En aquest cas el punt més alt és per k=5 i el segon és k=4, que és el que escollirem perquè ja ens havia sortit abans.

```
clusplot(train_data, train_data4$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



Podem observar que els 4 clusters que s'han format es van solapant l'un amb l'altre, amb el cluster 4 una mica més separat a la part alta.

Arbres de decisió.

En aquest apartat crearem un conjunt de regles que ens determinaran la probabilitat de sobreviure dels passatgers. Utilitzarem el dataset de train per construir el model i el de test per validar-lo.

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_data[-1]) i com a segon paràmetre el propi camp (train_data$Survived):

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

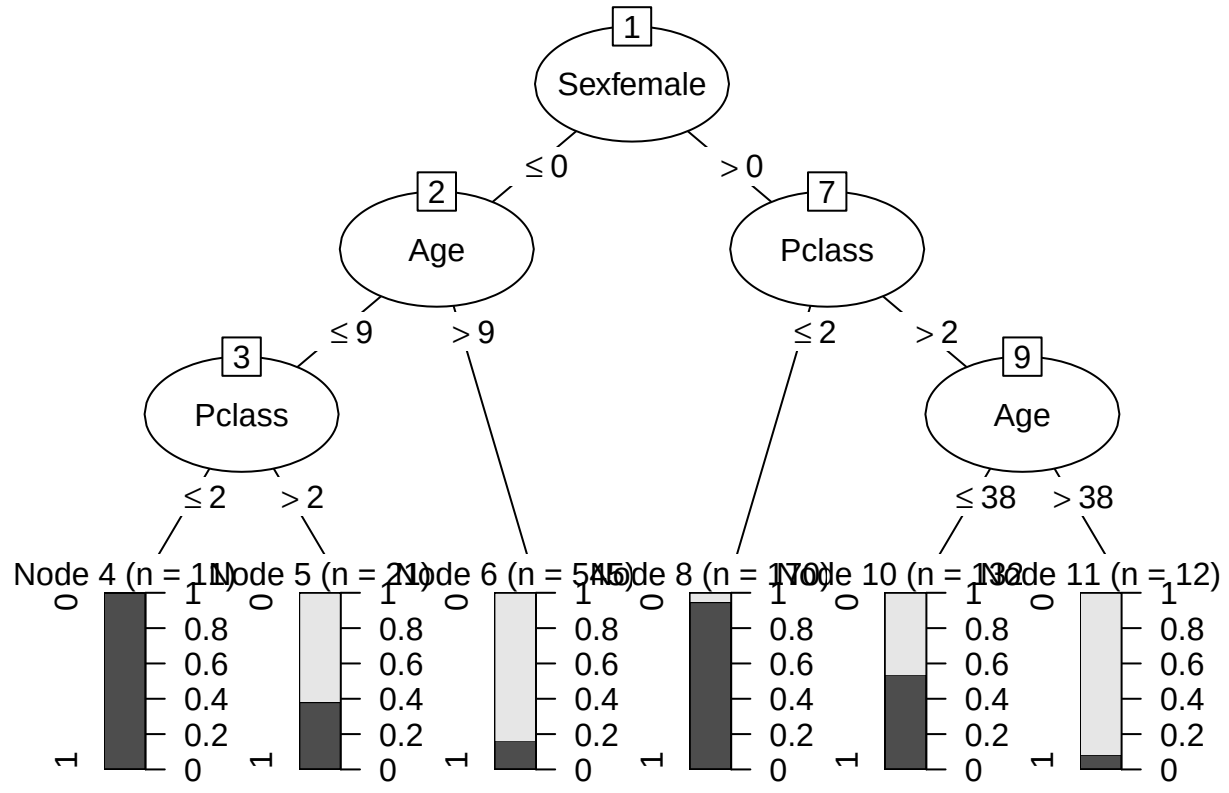
```
##
## Call:
## C5.0.default(x = train_data[-1], y = train_data$Survived)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon May 31 12:30:10 2021
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 891 cases (5 attributes) from undefined.data
##
## Decision tree:
##
## 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
```

Veiem que tenim 169 errors (un 19%), que indiquen el nombre de casos mal classificats.

La visualització de l'arbre obtingut és la següent:

```
plot(model)
```



En el gràfic podem veure esquemàticament els percentatges de supervivència en funció de les diferents variables. Ara descomposarem l'arbre en un set de regles amb el flag rules=TRUE:

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

```
##
## Call:
## C5.0.default(x = train_data[-1], y = train_data$Survived, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon May 31 12:30:10 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:

- Regla 1: Pclass > 2, Age $> 38 \rightarrow$ Survived $= 0$. Validesa: 90,6%
- Regla 2: Sexfemale $= 0 \rightarrow$ Survived 0. Validesa: 81,0%
- Regla 3: Pclass $<= 2$, Sexfemale $= 0$, Age $<= 9 \rightarrow$ Survived 1. Validesa: 92,3%
- Regla 4: Sexfemale $> 0 \rightarrow$ Survived 1. Validesa: 74,1%

En general podem concloure que els passatgers de classe 1 i 2 i les dones en concret tenen moltes probabilitats de sobreviure, així com els nens menors de 9 anys.

A continuació carreguem les dades de test i les utilitzarem per avaluar quants supervivents hi hauria fent servir el model creat.

```
testData <- read.csv('../data/test.csv',stringsAsFactors = FALSE)
dummies <- predict(dummyVars(~ Sex, data = testData), newdata = testData)
testData <- cbind(testData, dummies)
dim(testData)
```

```
## [1] 418  13
```

Veiem que les dades de test tenen 418 observacions i 13 variables, una menys que les d'entrenament ja que no hi ha el camp Survived, que l'intentarem predir ara:

```
predicted_model <- predict( model, testData, type="class" )
summary(predicted_model)
```

```
##   0   1
## 266 152
```

Segons la predicció del model, 266 passatgers no sobreviuen i 152 sí.

Model de regressió lineal múltiple (regresors quantitatius i qualitatius)

Proves de contrast d'hipòtesis, correlacions, regressions, etc.

Ara passem a avaluar la qualitat del primer model amb esbrinar si la regressió és lineal múltiple (amb regressors quantitatius i qualitatius). Carregem el conjunt de dades train_data per generar i esbrinar la qualitat del model amb la fòrmula de l'ajustament de regressió lineal lm().

```
# Torno als valors de train.csv, perquè considero que per fer el model de regressió la variable depender

trainData <- read.csv('../data/train.csv',stringsAsFactors = FALSE)
train_data <- trainData[ , c("Survived", "Pclass","Sex", "Age","Fare")]

# Considerem com a variable dependent, la variable Survived, la resta (Age, Sex, Fare i Pclass) les cons

model <- lm(Survived ~ Sex + Age + Pclass+Fare, train_data ) #Generació i valoració del model.
summary(model)
```

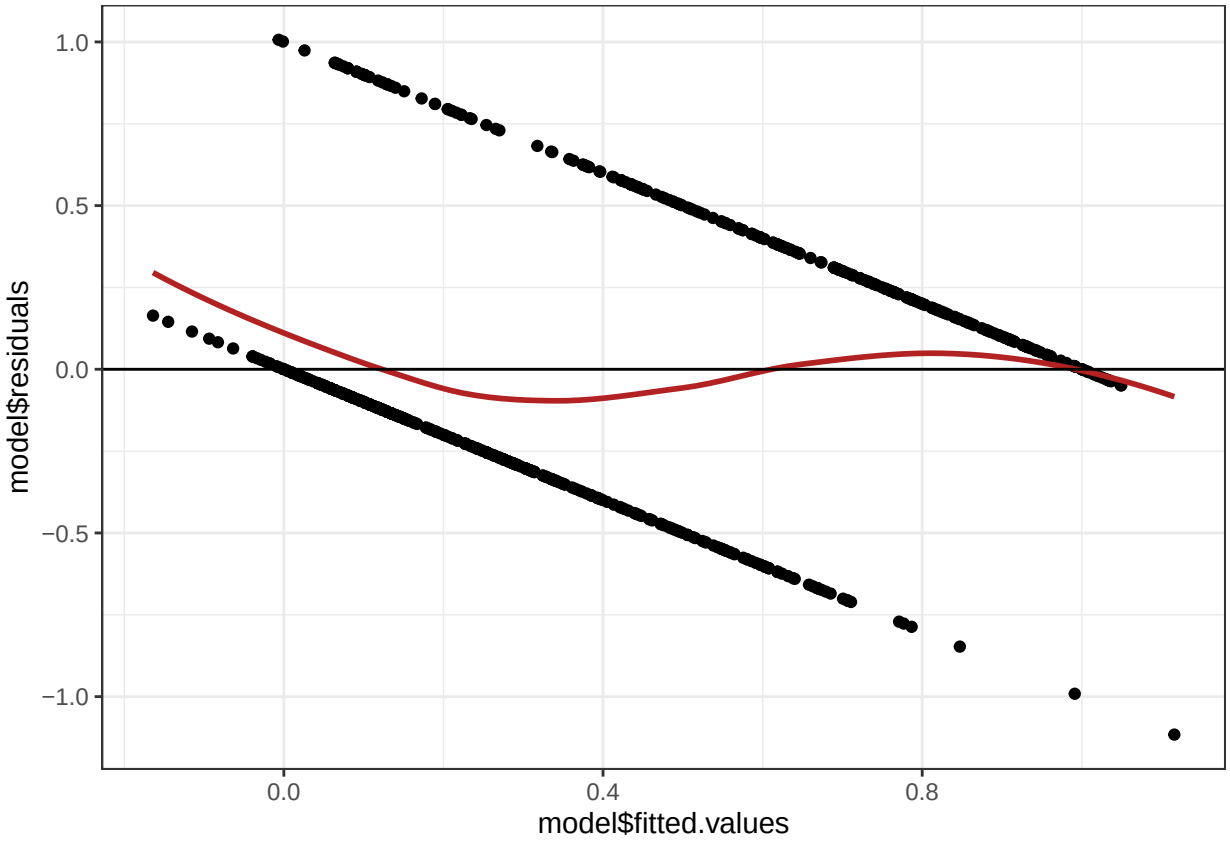
```
##
## Call:
## lm(formula = Survived ~ Sex + Age + Pclass + Fare, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11594 -0.25268 -0.06392  0.22965  1.00662
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.317e+00  7.699e-02  17.104  < 2e-16 ***
## Sexmale     -4.787e-01  3.084e-02 -15.518  < 2e-16 ***
## Age         -5.426e-03  1.091e-03  -4.975  8.2e-07 ***
## Pclass      -2.004e-01  2.250e-02  -8.907  < 2e-16 ***
## Fare         6.801e-05  3.321e-04   0.205    0.838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3849 on 709 degrees of freedom
##   (177 observations deleted due to missingness)
## Multiple R-squared:  0.3902, Adjusted R-squared:  0.3868
## F-statistic: 113.4 on 4 and 709 DF,  p-value: < 2.2e-16
```

Avaluació de la bondat de l'ajust, a partir del coeficient de determinació o R^2 (R^2 indica el grau d'ajust de la recta de regressió als valors de la mostra): a partir dels resultats anteriors amb la funció summary(), podem veure que el seu valor és Multiple R-squared: 0.3902. Amb aquest valor tant lluny de 1 no podem dir que hi ha regressió lineal entre les variables.

El model amb totes les variables introduïdes com a predictors té un R^2 (Multiple R-squared) 0.3902, el qual és capaç d'explicar el 39% de la variabilitat observada en la variable dependent "Survived". Com que no és molt proper al 100%, en principi no és un bon model. El p-value del model és significatiu (2.2e-16) perquè està molt per sota del 0.05 que és el valor d'alfa. Els asterics volen dir que tant la variable SexMale (els homes en aquest cas), com Pclass i Age són significatives per al resultat del model. La variable Fare no és significativa per al model.

A partir de la pregunta del principi, característiques dels passatgers que tenien més probabilitats de sobrevisure a l'enfonsament, el que podem deduir és que tant el Sexe, l'Edat i les condicions econòmiques en que es realitzaven el viatge (Classe i Tarifa) són rellevants i significatives per la supervivència.

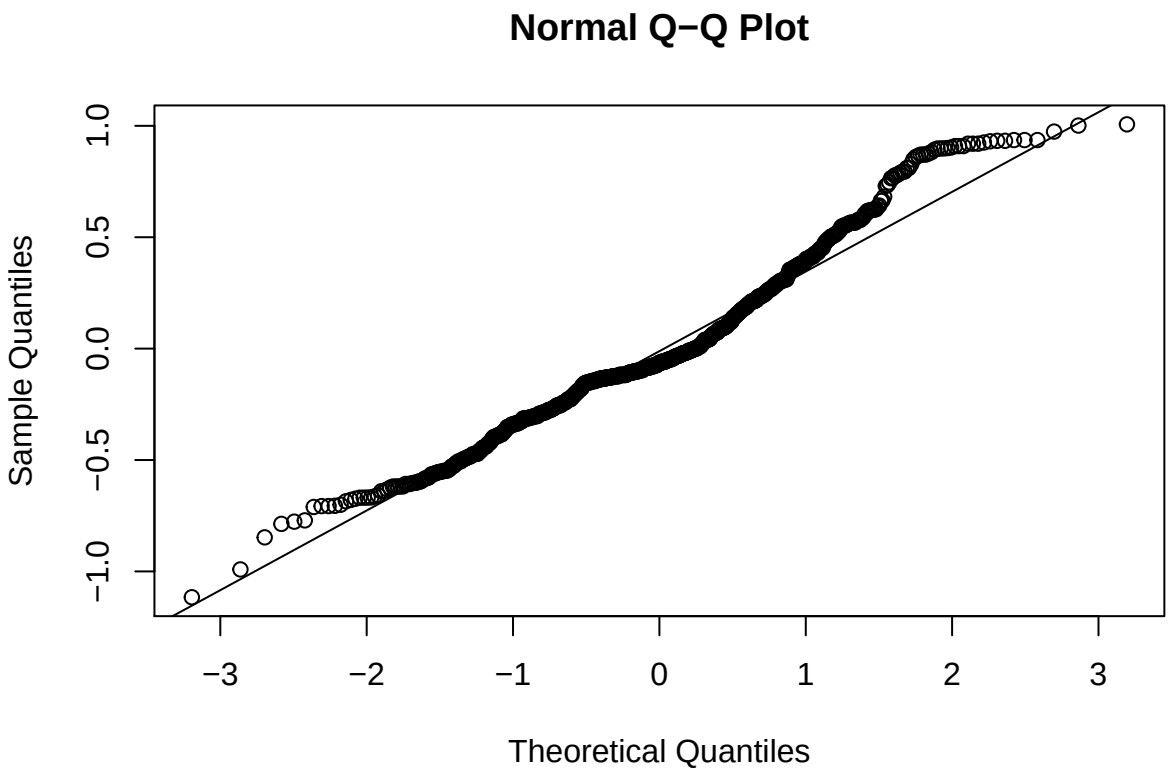
```
# Representació gràfica del model (valors ajustats enfront dels residus que ens permetrà veure si la va

ggplot(model,aes(model$fitted.values,model$residuals)) + geom_point() + geom_smooth(color = "firebrick"
```



Interpretació del resultat del gràfic:

El gràfic de dispersió serveix per validar la relació lineal entre la variable resposta ("Survived") i els predictors numèrics i categòrics (Age, Sex, Pclass i Fare). El gràfic mostra la dispersió entre cada un dels predictors i els residus del model. Com la relació no és lineal </>, els residus gairebé no es distribuexen al voltant de 0 amb una variabilitat més o menys constant al llarg de l'eix X. A més, el gràfic ens permet identificar dades atípiques per les corbes.

```
### Gràfic per visualitzar la normalitat.

# Gràfic quantil-quantil que compara els residus del model amb els valors d'una variable que es distribu

qqnorm(model$residuals)
qqline(model$residuals)
```



A partir del gràfic s'observa un patró de dispersió prou regular fins a l'extrem superior. A trets, sembla un patró aleatori dels residus. Això indica que no es compleix al 100% el supòsit de variància constant en els errors del model. D'altra banda el Q_Q plot, mostra que les dades no s'ajusten sempre a una normal.

Comprovació de la normalitat i homogeneïtat de la variància (homoscedasticitat).

Contrast d'hipòtesis:

Tenint en compte que la normalització redueix el biaix causat per la combinació de valors mesurats a diferents escales a l'hora d'ajustar-los a una escala comuna, típicament entre (-1,1) o entre (0,1), podríem dir que la nostra base de dades està normalitzada.

Per a realitzar el test partiriem d'una hipòtesis nul · la (variàcies iguals en les mostres) i una hipòtesis alternativa (variàcies iguals en les mostres).

H0 - Hipòtesis nul · la –> var1 = var2 H1 - Hipòtesis alternativa–> var1 <> var2

```
shapiro.test(trainData$Survived) # Test de Shapiro.
```

```
##
##  Shapiro-Wilk normality test
##
## data:  trainData$Survived
## W = 0.61666, p-value < 2.2e-16
```

Test de normalitat Shapiro on el p-valor (2.2e-16) resultant de la prova és més petit que el nivell de significació (0.05), això vol dir que s'observen diferències estadísticament significatives entre el grup de dades trainData per a la variable Survived.

Ara comprovarem l'homoscedasticitat amb el test de variança. Entre les proves més habituals hi ha el test de fligner.test, que s'aplica quan les dades segueixen una distribució normal.

```
library(car)

fligner.test(Survived ~ Age, data=trainData) # Test d'homogeneitat de les variables Survived i Age.
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Age
## Fligner-Killeen:med chi-squared = 73.115, df = 87, p-value = 0.8562
```

Com que la prova té un p-valor (0.8562) molt superior al nivell de significació (0.05), no es rebutja la hipòtesi nul · la d'homoscedasticitat i es conclou que la variable Age presenta variàncies estadísticament iguals o similars per als grups de Survived.

```
fligner.test(Survived ~ Sex, data=trainData) # Test d'homogeneitat de les variables Survived i Sex.
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Sex
## Fligner-Killeen:med chi-squared = 5.7729, df = 1, p-value = 0.01627
```

Com que la prova té un p-valor (0.01627), inferior al nivell de significació (0.05), es rebutja la hipòtesi nul · la d'homoscedasticitat i es conclou que la variable Sex no presenta variàncies estadísticament similars per als grups de Survived.

```
fligner.test(Survived ~ Pclass, data=trainData) # Test d'homogeneitat de les variables Survived i Pclass
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Pclass
## Fligner-Killeen:med chi-squared = 35.766, df = 2, p-value = 1.712e-08
```

Com que la prova té un p-valor (1.712e-08) molt inferior al nivell de significació (0.05), es rebutja la hipòtesi nul · la d'homoscedasticitat.

```
fligner.test(Survived ~ Fare, data=trainData) # Test d'homogeneitat de les variables Survived i Fare.
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Fare
## Fligner-Killeen:med chi-squared = 258.22, df = 247, p-value = 0.299
```

Com que la prova té un p-valor (0.299) superior al nivell de significació (0.05), no es rebutja la hipòtesi nul · la d'homoscedasticitat.

Correlacions:

```
corel<- cor(train_data$Survived,train_data$Fare, method = "pearson")
corel2<- cor(train_data$Survived,train_data$Pclass, method = "pearson")

train_data$Age[is.na(train_data$Age)] <- signif(mean(train_data$Age,na.rm=T), digits=2) # Per als valors
corel3<-cor(train_data$Survived,train_data$Age, method = "pearson")
```

Anàlisi de la relació entre les variables Survived i Fare. Aquesta informació és crítica a l'hora d'identificar quins poden ser els millors predictors per al model, quines variables presenten relacions de tipus no lineal (motiu pel que no poden ser incloses) i per identificar col · linealitat entre predictors. Fare seria un bon predictor. El valor és 0.2573065 perquè està entre 0 i 1 en valor positiu, com faig referència en el següent apartat.

Per poder realitzar comparacions entre variables, s'ha estandarditzar la covariància, generant els coeficients de correlació. He triat Pearson perquè funciona bé amb variables quantitatives que tenen una distribució normal, tot i que tots varien entre +1 (correlació positiva perfecta) i -1 (correlació negativa perfecta).

El coeficient de correlació entre Survived i Age és -0.0706572, i es tracta d'una correlació negativa perfecta.

Per últim el coeficient de correlació entre Survived i PClass és -0.338481, que també es tracta d'una correlació negativa perfecta.

Intervals de confiança del model:

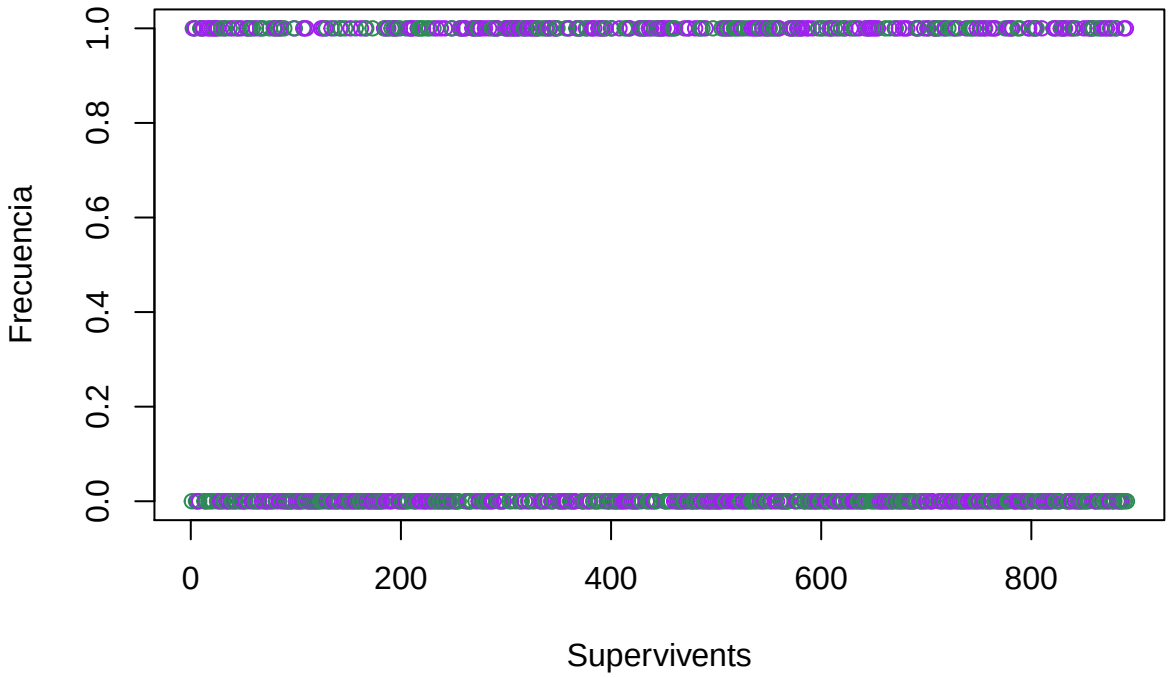
```
confint(model)
```

```
##                     2.5 %        97.5 %
## (Intercept)  1.1657399681  1.4680632744
## Sexmale     -0.5392228726 -0.4181063037
## Age         -0.0075668185 -0.0032843985
## Pclass      -0.2445946093 -0.1562399160
## Fare        -0.0005840531  0.0007200638
```

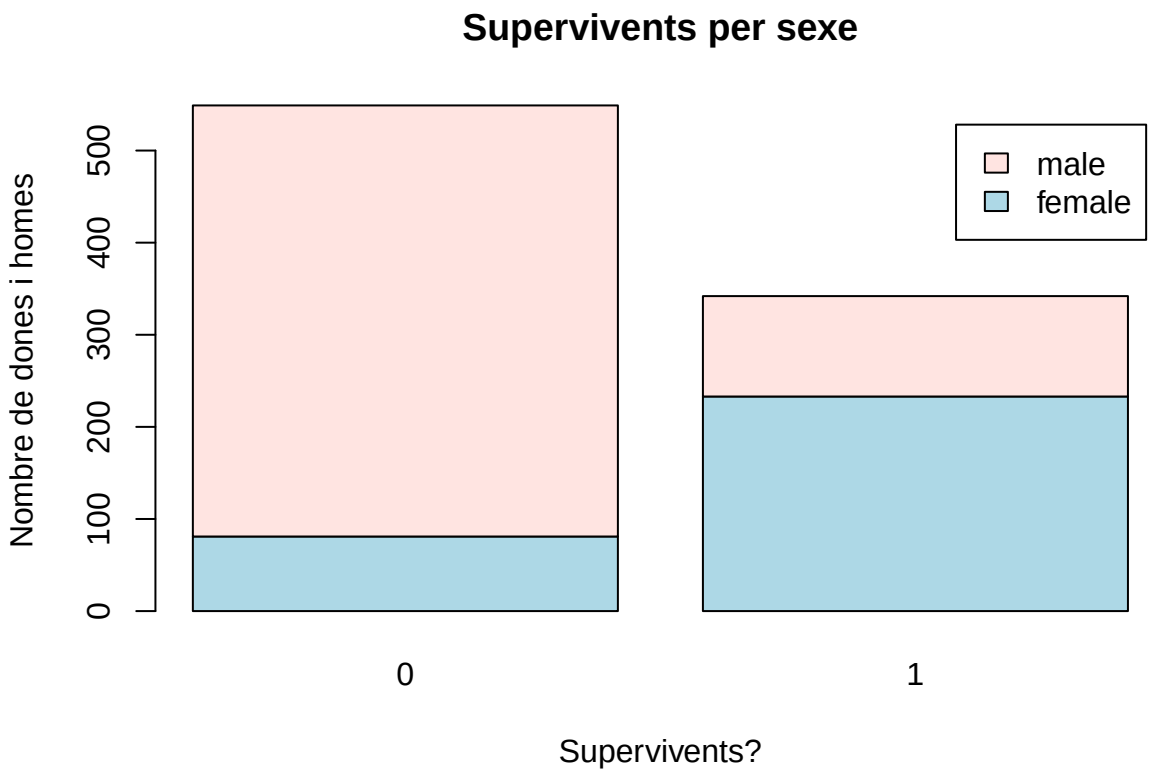
La funció confint mostra l'interval de confiança per cadascun dels coeficients parcials de regressió. Cadascun dels coeficients parcials de regressió dels predictors són les pendents d'un model de regressió lineal múltiple.

Exercici 5:

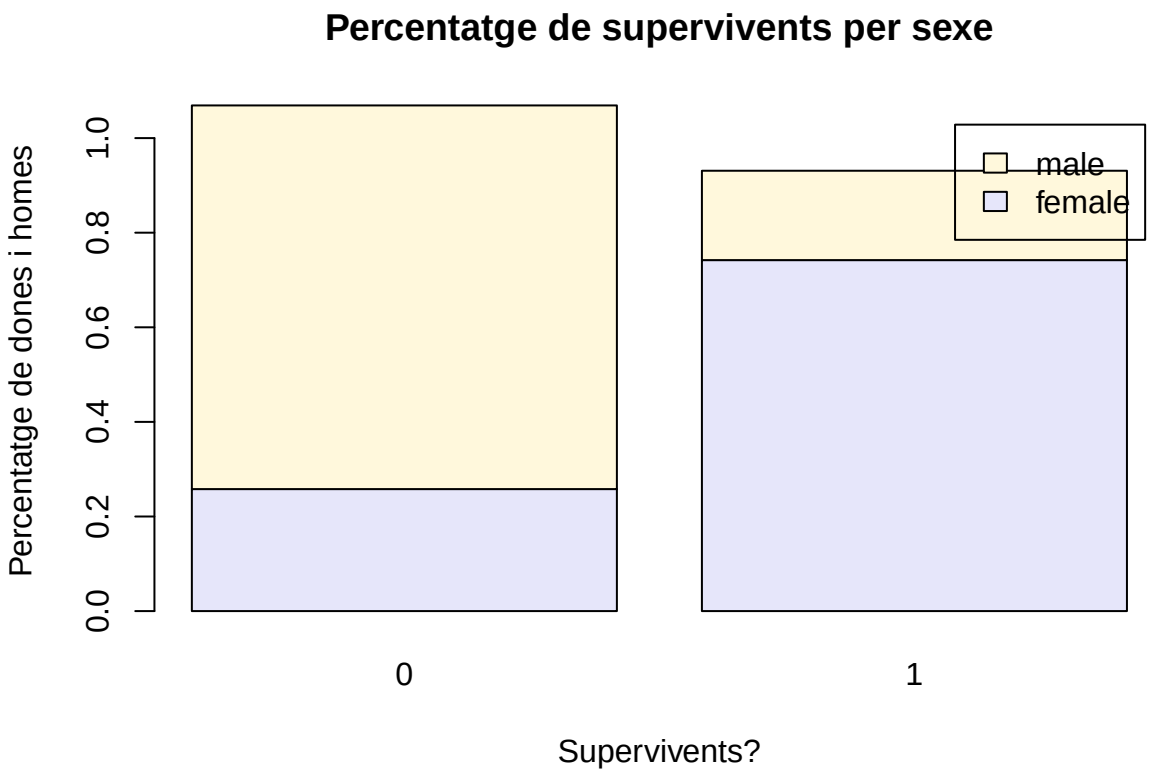
```
# Gràfic que mostra la distribució entre Supervivients i no supervivents.

plot(trainData$Survived,  xlab = "Supervivents", ylab = "Frecuencia", col = c("seagreen", "purple"))
```



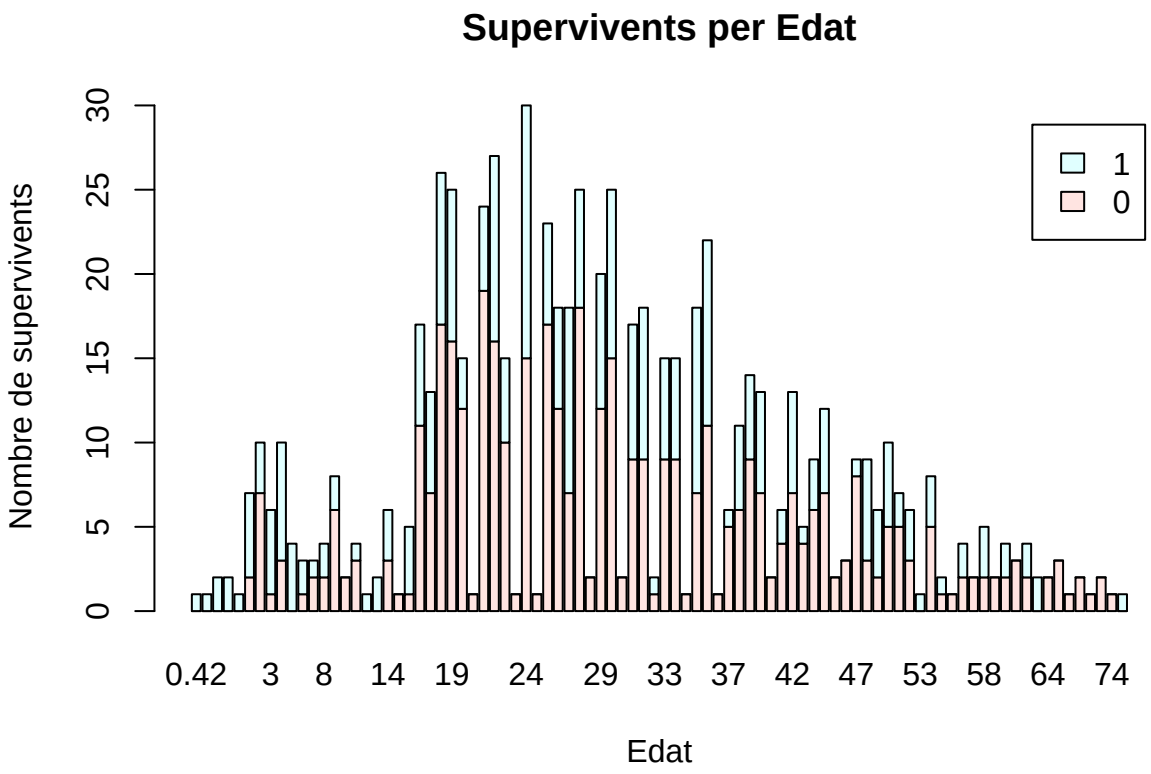

```
#Gràfics que representen les dades estudiades (Survived i Sex)

taula1<-table(trainData$Sex,trainData$Survived)

barplot(taula1,  col = c("lightblue", "mistyrose"), xlab="Supervivents?", ylab= "Nombre de dones i homes
```



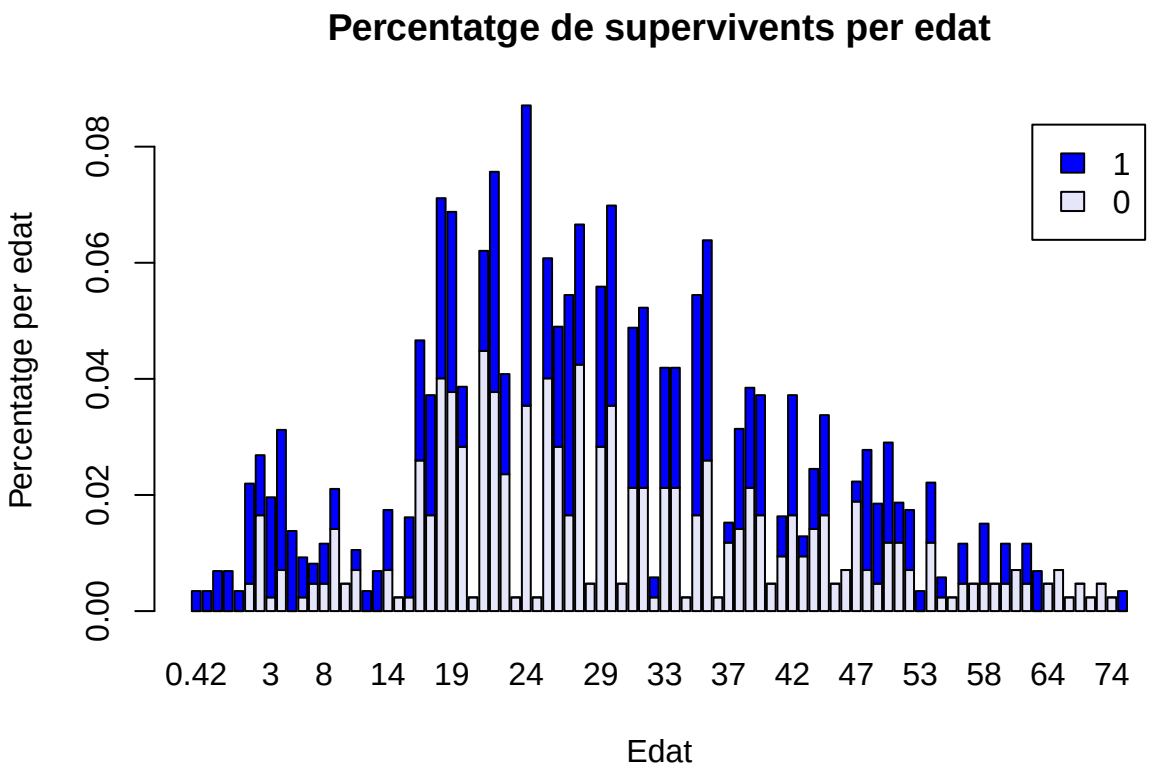

```
graf_prop_Sexe <- prop.table(taula1, margin = 1)
barplot(graf_prop_Sexe, col = c( "lavender", "cornsilk"), xlab="Supervivents?", ylab= "Percentatge de d
```



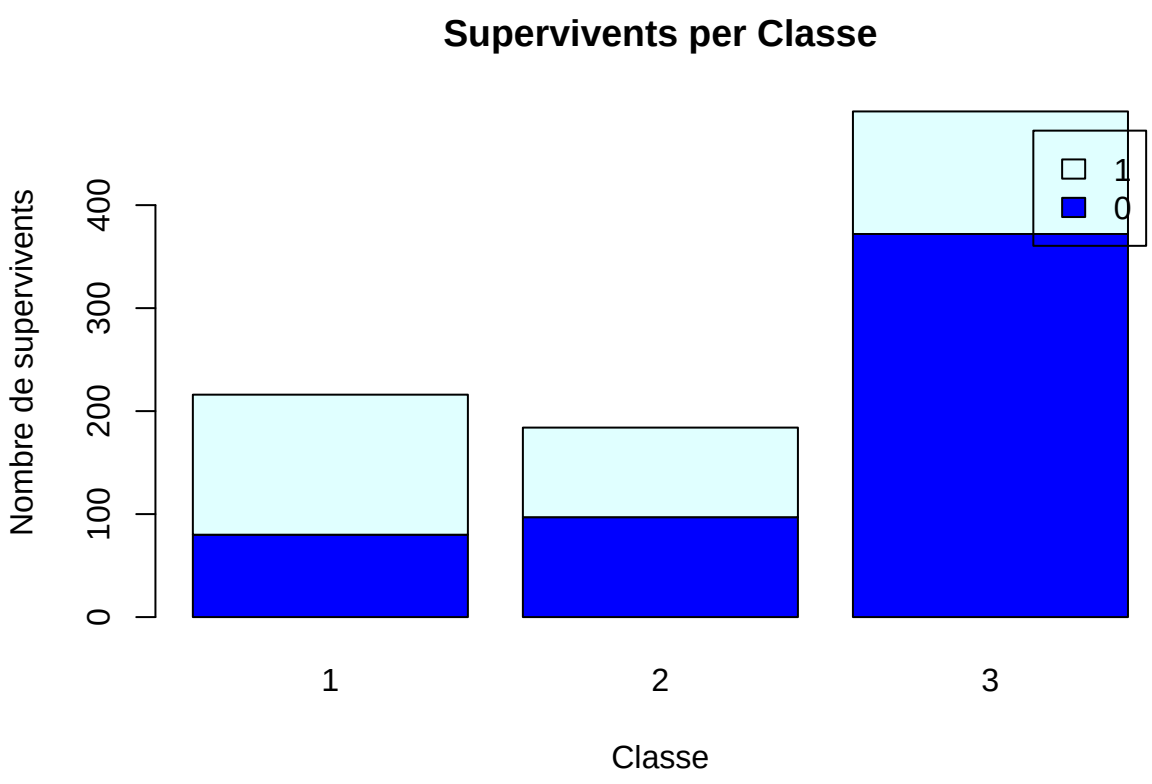

```
#Gràfics que representen les dades estudiades (Survived i Age):

taula2<-table(trainData$Survived,trainData$Age)

barplot(taula2,  col = c("mistyrose", "lightcyan"), xlab="Edat", ylab=" Nombre de supervivents", legend
```



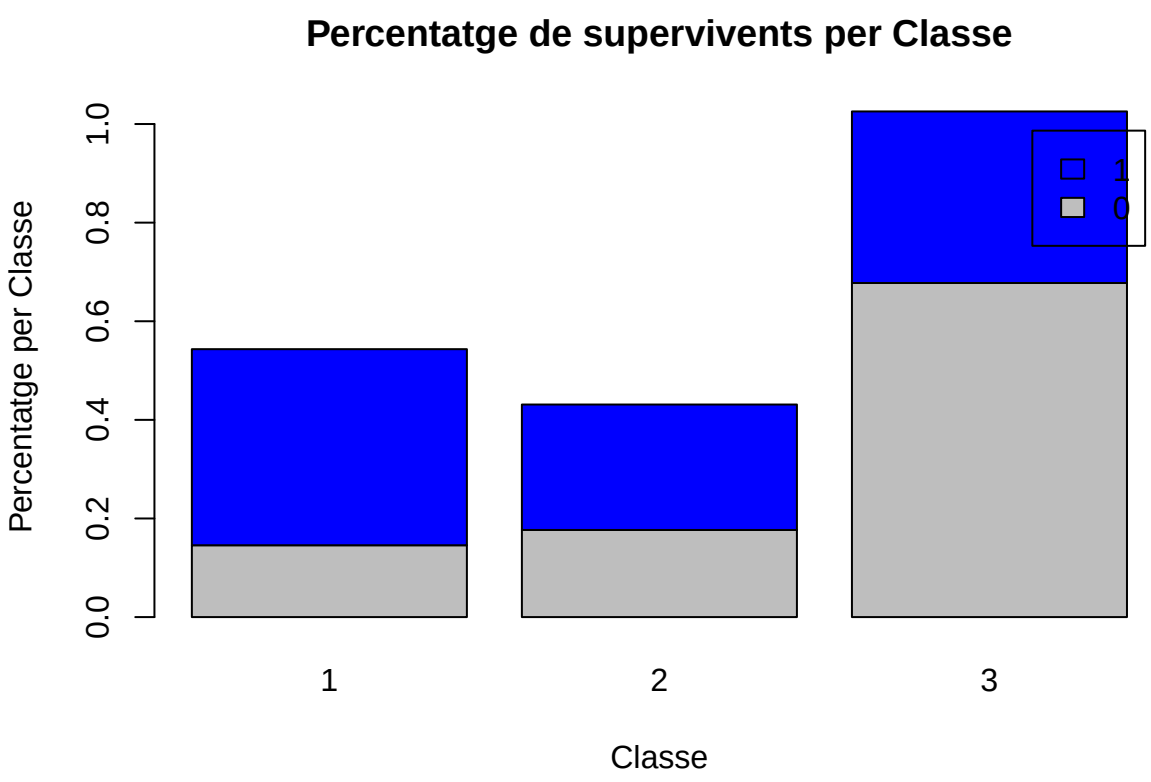

```
graf_prop_Edat <- prop.table(taula2, margin = 1)
barplot(graf_prop_Edat, col = c( "lavender", "blue"), xlab="Edat", ylab= "Percentatge per edat", legend
```




```
#Gràfics que representen les dades estudiades (Survived i Pclass):

taula3<-table(trainData$Survived,trainData$Pclass)

barplot(taula3,  col = c("blue", "lightcyan"), xlab="Classe", ylab=" Nombre de supervivents", legend=TRU
```




```
graf_prop_Classe <- prop.table(taula3, margin = 1)
barplot(graf_prop_Classe, col = c( "grey", "blue"), xlab="Classe", ylab= "Percentatge per Classe", leger
```



***** # Exercici 6: *****

- Clustering: és difícil extreure conclusions del model d'agregació creat ja que els clusters obtinguts no mostren una distinció clara entre ells sinó que queden agrupats des de la part baixa del gràfic fins a la part més alta en grups semblants que se solapen entre ells.
- Arbres de decisions amb regles: en aquest cas sí que podem obtenir una predicció dels supervivents utilitzant el model creat. Primer hem generat un arbre de decisió que ens porta a saber la probabilitat de supervivència en base a les variables utilitzades amb un error del 19%. A partir de l'arbre obtenim el joc de regles que també ens indiquen la probabilitat de supervivència amb el percentatge de validesa per cada regla. Per últim calculem la predicció utilitzant el joc de dades de test.
- Model de regressió: amb el model de regressió s'avalua la qualitat del primer model esbrinant si la regressió és lineal múltiple (amb regressors quantitatius i qualitatius) Es realitza amb el conjunt de dades train_data per esbrinar la qualitat del model amb la fòrmula de l'ajustament de regressió lineal lm(). S'avalua la bondat de l'ajust, a partir del coeficient de determinació o Rˆ2. Es conclou amb els valors de l'anàlisis que, en principi, el model no és prou bo. Realitzem també un gràfic per fer una interpretació visual del model de regressió, el qual serveix per validar la relació lineal entre la variable resposta ("Survived") i els predictors numèrics i categòrics (Age, Sex, Pclass i Fare). El gràfic mostra la dispersió entre cada un dels predictors i els residus del model.
- S'han calculat contrast d'hipòtesis, correlacions i regressions arribant a les següents conclusions:

Per realitzar el test s'ha partit de l'hipòtesis nul · la (variàcies iguals en les mostres) i l'hipòtesis alternativa (variàcies iguals en les mostres).

Correlacions (només es poden realitzar amb variables numèriques): el millor valor predictor és Fare perquè és un valor positiu que està entre 0 i 1. Age i Pclass són predictors a tenir en compte, però saben que són coeficients de correlació negatius perfectes.

La funció confint mostra l'interval de confiança per cadascun dels coeficients parcials de regressió. Cadascun dels coeficients parcials de regressió dels predictors són les pendents d'un model de regressió lineal múltiple.

- Comprovació de la normalitat i homogeneïtat de la variància (homoscedasticitat): hem comprovat que la base de dades utilitzada està normalitzada utilitzant el test de normalitat de Shapiro. L'homoscedasticitat l'hem comprovada amb un test de variança anomenat fligner.test que hem considerat el més adequat per ser d'ús habitual quan les dades segueixen una distribució normal. També s'han realitzat dos gràfics de comprovació de la normalitat i la homohomoscedasticitat.

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
```