Titanic - Learning from the Disaster

Jose Antonio Montero, Mireia Mora

04/01/2021

Introducció

El Titanic va ser el més gran vaixell de passatgers del món que després d'impactar amb un iceberg, es va enfonsar durant la nit del 14 i la matinada del 15 d'abril de 1912 durant el viatge inaugural que es va fer desde Southampton cap a Nova York.

1502 persones de les 2224 que hi viatjaven van morir, i de tot aquest volum de gent hi havien grups de persones que tenien més probabilitats de sobreviure que unes altres.

En aquesta pràctica utilitzarem les dades dels passatgers per tal de crear models que ens indiquin quines persones tenien més probabilitats de sobreviure.

1.Descripció del Dataset.

¿Per què és important i quin/es problema/preguntes preten respondre?

El dataset del Titanic és un dels més popular en l'anàlisis de dades per l'impacte que va tenir el succés. Principalment està orientat a **calcular la probabilitat de supervivència dels passatgers en funció de les seves característiques (edat, clase social, preu del passatge, familia, etc)** però també dona lloc a fer-se preguntes de l'estil **si el preu del passatge dels que sobreviuen és més gran que el preu del passatge dels que moren**, per tant orientarem els nostres anàlisis per a respondre aquest tipus de preguntes.

Lectura de dades.

En primer lloc, llegiu el fitxer de dades i verifiqueu que els tipus de dades són interpretats correctament. Si s'escau, feu les conversions de tipus que siguin oportunes.

- Llegirem els datasets "test" i "gender", i els combinarem en un únic fitxer a través de la columna PassengerId
- Llegirem el dataset "train", i tindrem ja carregats tots dos fitxers per tal de realitzar les operacions.

Farem crides a les funcions per tots dos fitxers de dades.

```
library(readr)
library(rmarkdown)
library(knitr)

## Selecció del directori de treball on es troben els documents que volem
setwd("C:/Users/mirei/OneDrive/Documentos")
#setwd("D:/UOC_ML/S1-Tipologia i Cicle Vida Dades/PAC/PRAC2")

# Llegim els datasets de test i gender per tal de realitzar una fusió posterior
#entre els dos fitxers
titanic_test <- read.csv("titanic_test.csv", dec=".", stringsAsFactors = FALSE)
```

```r
titanic_gender<- read.csv("gender_submission.csv")

# Fusionem els dos fitxers tenint en compte la variable PassengerId i l'ordre
#de les capçaleres de les columnes
titanic_test = merge(x=titanic_gender, y=titanic_test, by=c("PassengerId"))

# Llegim el dataset de train
titanic <- read.csv("titanic.csv", dec=".", stringsAsFactors = FALSE)
```

En aquest moment ja tenim tots dos fitxers amb les variables per analitzar.

- PassengerId : Codi d'identificació del passatger
- Survived : Supervivencia (0=No, 1=Sí)
- PClass : Classe, estatus del passatger (1=1ª, 2=2ª, 3=3ª)
- Name : Nom del passatger
- Sex : Sexe del passatger
- Age : Edat
- SibSp : número de germans/conjugues a bord del vaixell
- Parch : número de pares/fills a bord del vaixell
- Ticket : Número del ticket
- Fare : Tarifa del passatge
- Cabin : Número de cabina
- Embarked : Port d'embarcament (C=Cherburgo, Q=Queenstown, S=Southampton)

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

En aquest apartat realitzarem la integració i l'exploració de dades fent crides a les funcions que crearem per tots dos fitxers, per tal de crear una estructura de dades coherent i única que contingui més quantitat d'informació, i relacionant atributs que ens permetin prescindir d'informació més redundant.Així mateix crearem noves variables a través d'unes altres ja existents. En aquest apartat i resumint, realitzarem el següent :

- Discretitzarem la variable Sex factoritzant-la on female=1 i male=0
- Discretitzarem la variable Embarqued factoritzant-la on el port de Southampton (S=0), el port de Cherburgo (C=1) i el port de Queenstown (Q=2).
- Crearem nova variable "Titol" que contingui el titol (Mr, Mrs, Ms...) i la discretitzarem també per tal de poder analitzar posteriorment la probabilitat de supervivència sobre un titol o un altre.
- Crearem una nova variable "Familysize" sumant les columnes Sibsp i Parch que ens dirà el nombre de familiars que viatjaven amb un passatger
- Crearem una nova variable "Isalone" mirant els valors que conté la columna "Familysize", si és superior a 0 vol dir que el passatger anava acompanyat, si és 0 vol dir que el passatger anava sol.
- Eliminar la variable "Name", un cop creada i degudament omplerta la variable "Titol", ja que no ens aportarà res per l'anàlisi
- Eliminar les variables "Parch, SibSp, i FamilySize", un cop creada i omplerta la variable "Isalone".

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)

# Funció que converteix la columna amb la variable Sex i la convertim a Integer
canvi_sex <- function(x)
{
   # Assignem el valor 1 a la columna Sex quan es trobi que el contingut és female
   x[x$Sex=="female","Sex"] = 1
   x[x$Sex=="male","Sex"] = 0
   x$Sex<-as.integer(x$Sex)
  return(x)
}

# cridem a la funció canvi_sex
titanic_test <- canvi_sex(titanic_test)
titanic <- canvi_sex(titanic)

str(titanic)
```

```
## '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        : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ 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" ...
```

```
# Funció que converteix la columna amb la variable Embarked depenent del port
#on hagi embarcat i la Factoritzem
canvi_port <- function(x)
{

   x[x$Embarked=="S","Embarked"] = "0"
   x[x$Embarked=="C","Embarked"] = "1"
   x[x$Embarked=="Q","Embarked"] = "2"

   x$Embarked <-as.factor(x$Embarked)
  return(x)
}

# cridem a la funció canvi_port
titanic_test <- canvi_port(titanic_test)
titanic <- canvi_port(titanic)

str(titanic)
```

```
## '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        : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ 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   : Factor w/ 4 levels "","0","1","2": 2 3 2 2 2 4 2 2 2 3 ...
```

```
# Creem la columna Titol amb valors NA inicialment
titanic_test <- mutate(titanic_test, Titol="NA")
titanic <- mutate(titanic, Titol="NA")

# Funció que afegeix a la nova columna creada el titol de tractament que trobi
#a la columna Name del dataset
assigna_titol <- function(x)
{
  # Creem un dataset amb les possibles titols de presentació de les persones
  titulitis <- c("Master","Miss","Mr","Mrs","Dr","Rev","Mlle","Ms","Mme","Don",
```

```
                "Major","Col","Capt","Countess","Jonkheer")

  titulitis_c <- str_c(titulitis,collapse = "|")
  x$Titol=str_extract(x$Name,titulitis_c)

  return(x)
}

# cridem a la funció Assigna_titol
titanic_test <- assigna_titol(titanic_test)
titanic <- assigna_titol(titanic)

str (titanic)
```

```
## 'data.frame':    891 obs. of  13 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        : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ 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   : Factor w/ 4 levels "","0","1","2": 2 3 2 2 2 4 2 2 2 3 ...
##  $ Titol      : chr  "Mr" "Mr" "Miss" "Mr" ...
```

```
# Funció que converteix el titol de tractament a la nova columna de string a
#numèric i després el factoritzem
canvi_titol <- function(x)
{
  x[x$Titol=="Mr","Titol"] = "1"
  x[x$Titol=="Don","Titol"] = "1"
  x[x$Titol=="Miss","Titol"] = "2"
  x[x$Titol=="Mlle","Titol"] = "2"
  x[x$Titol=="Ms","Titol"] = "2"
  x[x$Titol=="Mrs","Titol"] = "3"
  x[x$Titol=="Mme","Titol"] = "3"
  x[x$Titol=="Master","Titol"] = "4"
  x[x$Titol=="Dr","Titol"] = "5"
  x[x$Titol=="Rev","Titol"] = "5"
  x[x$Titol=="Major","Titol"] = "5"
  x[x$Titol=="Col","Titol"] = "5"
  x[x$Titol=="Capt","Titol"] = "5"
  x[x$Titol=="Countess","Titol"] = "5"
  x[x$Titol=="Jonkheer","Titol"] = "5"

  x$Titol<-as.factor(x$Titol)
  return(x)
}

titanic_test <- canvi_titol(titanic_test)
```

```r
titanic <- canvi_titol(titanic)

str (titanic)
```

```
## 'data.frame':    891 obs. of  13 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        : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ 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   : Factor w/ 4 levels "","0","1","2": 2 3 2 2 2 4 2 2 2 3 ...
##  $ Titol      : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 1 1 1 4 1 1 ...
```

```r
# creació de la nova columna Familysize amb atributs de de les columnes SibSp i Parch
titanic_test <- mutate(titanic_test, FamilySize=titanic_test$SibSp+titanic_test$Parch)
titanic <- mutate(titanic, FamilySize=titanic$SibSp+titanic$Parch)

str (titanic)
```

```
## 'data.frame':    891 obs. of  14 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        : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ 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   : Factor w/ 4 levels "","0","1","2": 2 3 2 2 2 4 2 2 2 3 ...
##  $ Titol      : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 1 1 1 4 1 1 ...
##  $ FamilySize : int  1 1 0 1 0 0 0 4 2 1 ...
```

```r
# eliminem la columna Name ja que ja tenim creada i discretitzada la columna titol
titanic_test$Name<-NULL
titanic$Name<-NULL

#creem nova variable Isalone on comprovarà si la columna Familysize es major
#que 0, si és així el passatger no era ell sol, sino que tenia més familiars,
#en canvi, si el valor era 0 vol dir que el passatger no tenia a ningú més a bord
titanic_test<-mutate(titanic_test, Isalone=ifelse(titanic_test$FamilySize>0,1,0))
titanic<-mutate(titanic, Isalone=ifelse(titanic$FamilySize>0,1,0))

# eliminem les columnes Parch, Sibsp i Familysize ja que tenim creada i
#discretitzada la columna Isalone
titanic_test$Parch<-NULL
titanic_test$SibSp<-NULL
```

```
titanic_test$FamilySize<-NULL

titanic$Parch<-NULL
titanic$SibSp<-NULL
titanic$FamilySize<-NULL
```

3. Neteja de dades.

3.1 Reducció.

Dintre de la neteja de dades una de les opcions és la reducció que consisteix a eliminar variables redundants o que no estan relacionades amb el fet que es vol analitzar.

```
library(knitr)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.4     v purrr   0.3.4
## v tidyr   1.1.2     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
library(DescTools)
```

```
##
## Attaching package: 'DescTools'
```

```
## The following objects are masked from 'package:caret':
##
##     MAE, RMSE
```

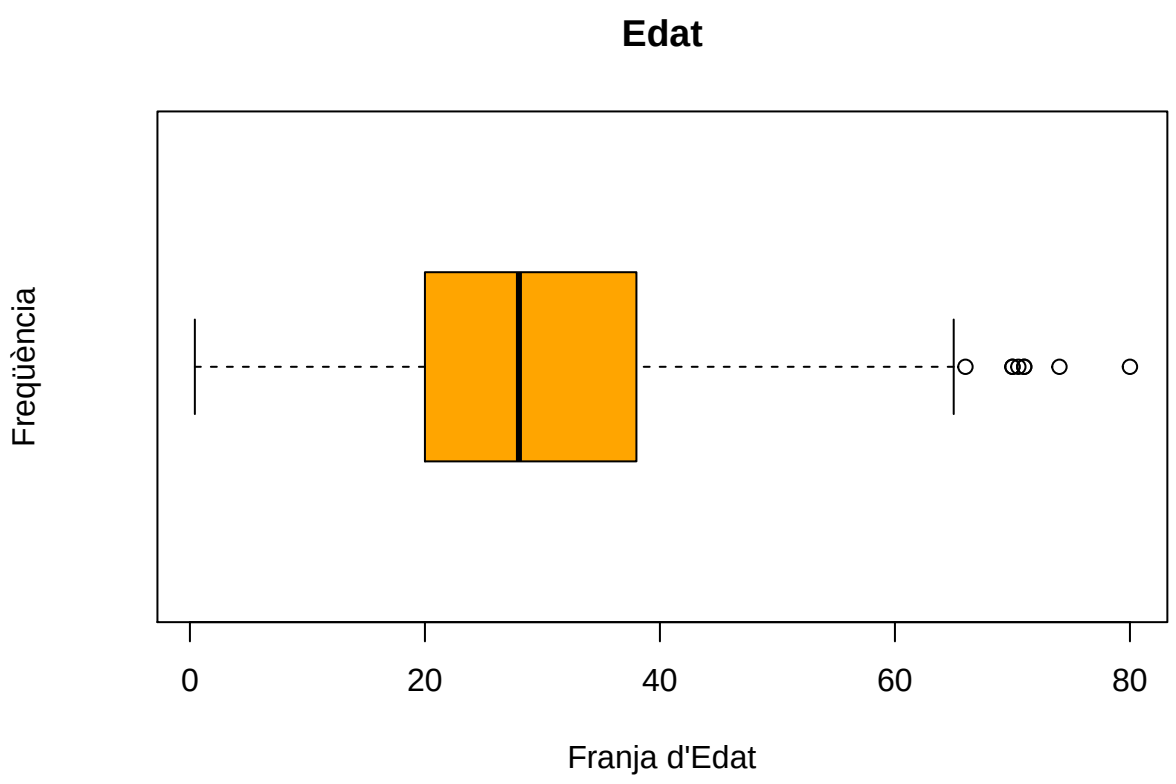
```
##Esborrarem les variables Cabin,Ticket,Name,PassengerId
##Una vegada tiguem calculada la IsAlone també esborrarem Parch,SibSp i FamilySize
titanic = titanic[,!(names(titanic) %in% c("Ticket","Cabin","Name","PassengerId"))]
titanic_test = titanic_test[,!(names(titanic_test) %in% c("Ticket","Cabin","Name","PassengerId"))]
```

3.2 Identificació i tractament de valors extrems.

Tractament de outliers. En aquest apartat buscarem aquelles dades que es troben molt allunyades de la distribució normal de les variables que utilitzarem, és a dir, les desviacions d'aquells valors que representarem mitjançant gràfics boxplot de diferents variables: age, fare, pclass. Amb aquestes tres variables farem combinacions per tal de veure els valors extrems de :

- Edat : buscarem aquells valors extrems per la variable de l'edat dels passatgers.
- Tarifa : buscarem aquells valors amb desviacions sobre la tarifa agafada per cada passatge.
- Tarifa per Classe : buscarem aquells valors extrems segons la tarifa del passatger segons la classe adquirida del passatge (1^a, 2^a o 3^a classe).
- Edat per Classe : buscarem aquells valors extrems de l'edat dels passatgers segons la classe ocupada al passatge del Titanic (1^a, 2^a o 3^a classe).

```
titanic.bp<-boxplot(titanic$Age, horizontal = TRUE, col=c("orange"), main="Edat",
                    xlab="Franja d'Edat", ylab="Freqüència")
```




```
summary(titanic$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

```
titanic.bp$out
```

```
## [1] 66.0 71.0 70.5 71.0 80.0 70.0 70.0 74.0
```

```
titanic.bp$stats
```

```
##       [,1]
## [1,]  0.42
## [2,] 20.00
## [3,] 28.00
## [4,] 38.00
## [5,] 65.00
```

El valor màxim respecte a la variable de l'edat és de 80 anys, mentre que la mitja d'edat és de 29,7 anys. El 50% de les vegades l'edat ha estat de 28 anys o inferior. Els outliers serien el resultat d'aquells valors que disten 3 cops el rang IQR por sota de Q1 o per sobre de Q3.

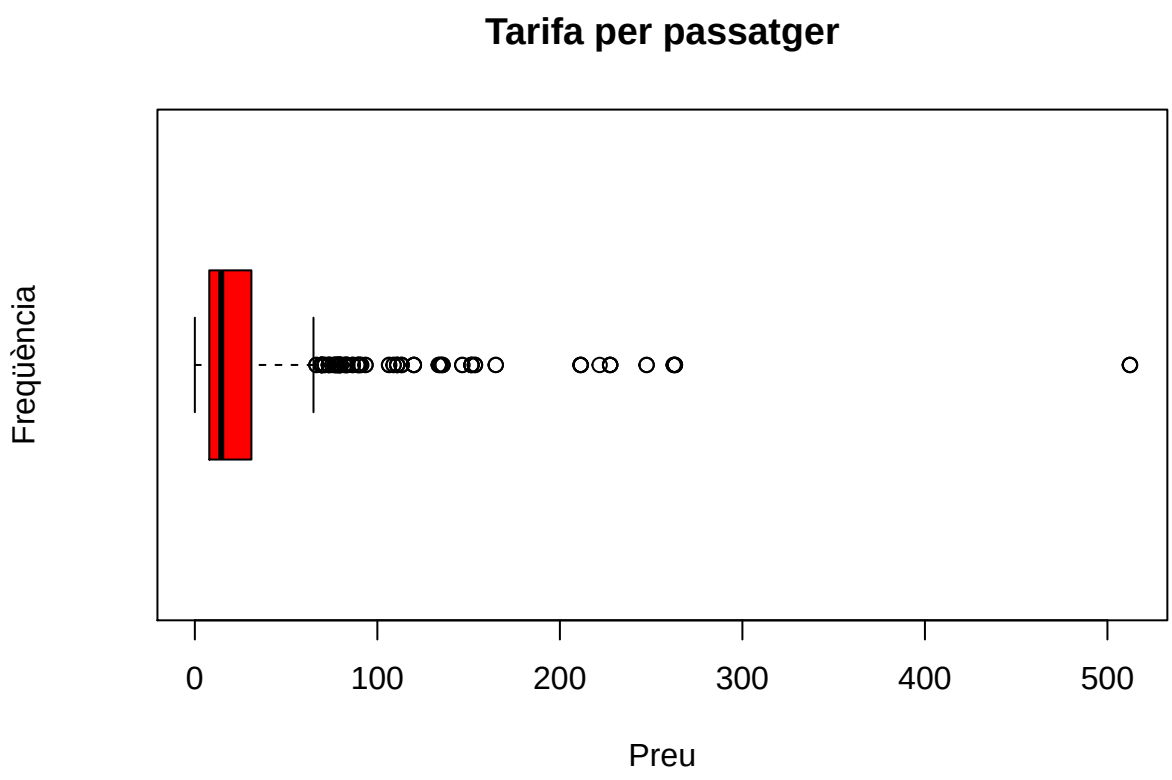
Calcularíem primer el rang interquartílic (estimació estadística de la dispersií d'una distribució de dades) IQR = Q3 – Q1 = 38 - 20.12 = 17,88

Els valors extrems del boxplot de l'edat dels passatgers serien, tots aquells valors que superin el mostreig de la tarifa de 64,82 anys, calculada a través de la següent fórmula:

Q3 + 1,5 * IQR # 38 + 1,5 * 17,88 = 64,82

Llistem amb la sentència titanic.bp$out els 8 valors extrems superiors a 64,82 anys que trobem al dataset train. Veiem en aquest punt que hi ha molts valors NA que en els propers apartats es solventarà.

```
boxplot(titanic$Fare,horizontal =TRUE, col=c("red"), main="Tarifa per passatger",
        xlab="Preu", ylab="Freqüència")
```




```
summary(titanic$Fare)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    7.91   14.45   32.20   31.00  512.33
```

El preu màxim de la tarifa adquirida és de 512,33, mentre que la mitja és de 32,20. El 50% de les vegades el preu ha estat de 14,45 o inferior. Aquesta mitjana està molt allunyada de la mitja, que seria una mica superior al doble de la mitjana.

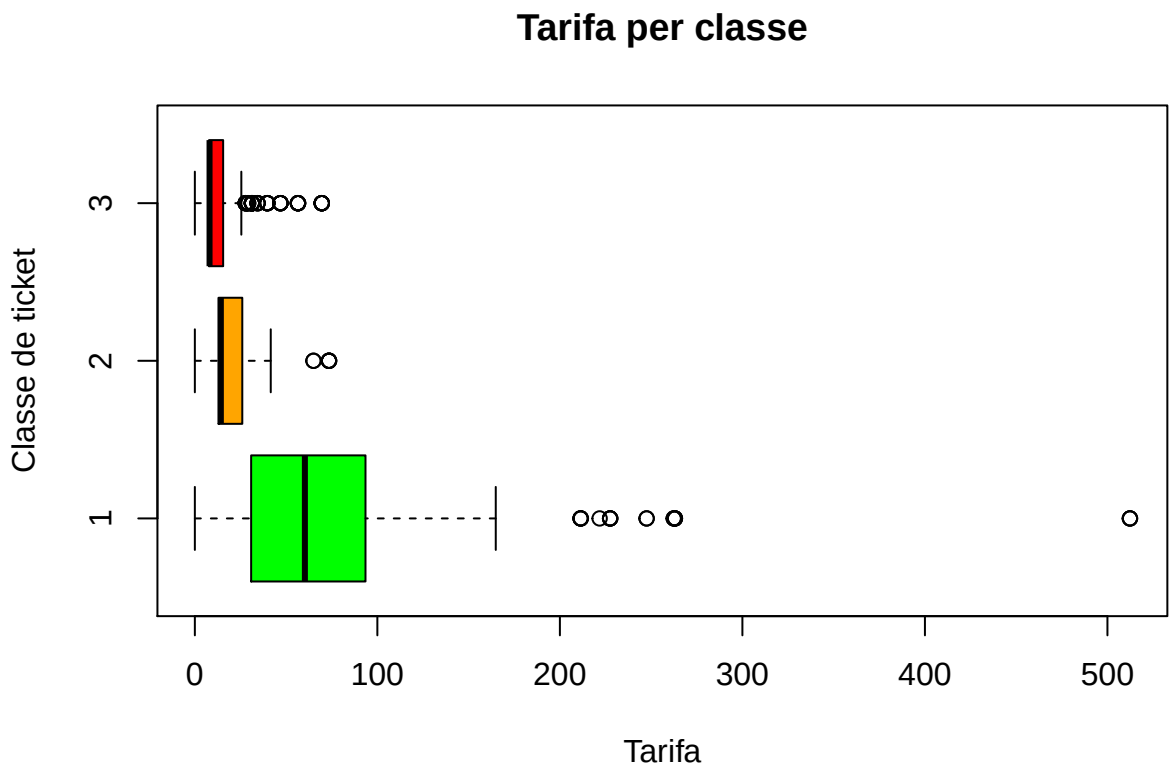
En aquest cas el rang IQR seria: IQR = Q3 – Q1 = 31 - 7.91 = 23,09

Els valors extrems del boxplot de les tarifes dels passatgers serien, tots aquells valors que superin el mostreig de la tarifa de 65,63, calculada a través de la següent fórmula:

Q3 + 1,5 * IQR # 31 + 1,5 * 23,09 = 65,63

Com es veu en el gràfic trobem molts outliers superiors a aquest valor. No hi ha NA en aquesta variable, tots els valors es poden considerar vàlids per ser analitzats.

```
titanic.bp<-boxplot(titanic$Fare~titanic$Pclass, horizontal = TRUE,
                    col=c("green","orange","red"), main="Tarifa per classe",
                    xlab = "Tarifa", ylab = "Classe de ticket")
```

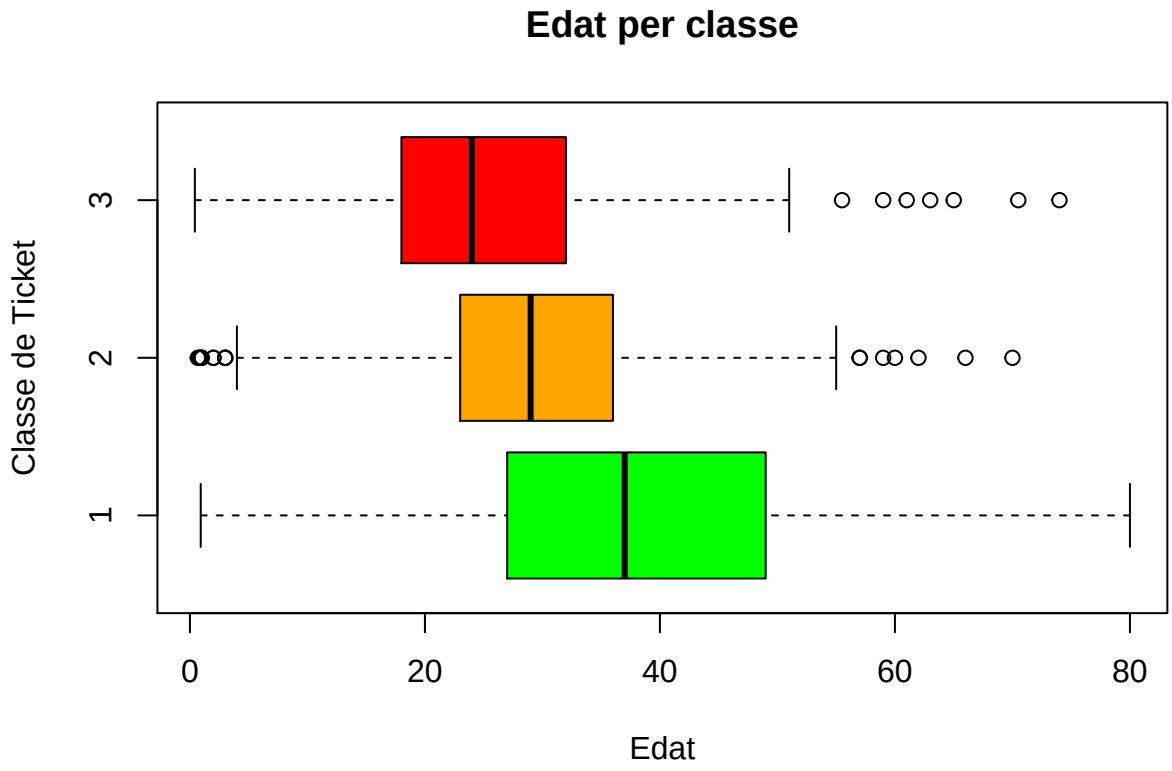



```
titanic.bp$out
```

```
##  [1] 263.0000 263.0000 247.5208 512.3292 247.5208 262.3750 263.0000 211.5000
##  [9] 227.5250 263.0000 221.7792 227.5250 512.3292 211.3375 227.5250 227.5250
## [17] 211.3375 512.3292 262.3750 211.3375  73.5000  73.5000  73.5000  65.0000
## [25]  73.5000  73.5000  65.0000  31.2750  29.1250  31.3875  39.6875  46.9000
## [33]  27.9000  46.9000  56.4958  34.3750  31.2750  34.3750  69.5500  39.6875
## [41]  27.9000  56.4958  29.1250  69.5500  31.3875  69.5500  31.3875  31.3875
## [49]  39.6875  29.1250  69.5500  27.9000  46.9000  34.3750  46.9000  56.4958
## [57]  31.2750  31.2750  31.2750  27.9000  39.6875  27.9000  56.4958  46.9000
## [65]  46.9000  39.6875  56.4958  34.3750  29.1250  69.5500  31.2750  27.9000
## [73]  39.6875  56.4958  56.4958  69.5500  31.2750  69.5500  29.1250
```

En aquest gràfic podem observar valors extrems a les tarifes de totes les classes. Observem que el valors extrems de tercera classe, comencen en una tarifa molt inferior als valors extrems de les tarifes de primera classe. Els valors extrems del preu de la tarifa augmenten la seva tarifa a mida que puja la classe d'estada del passatger al vaixell, d'aquesta manera, els passatgers de primera classe tenen valors extrems de tarifes de billet superiors, i els passatgers de tercera classe tenen valors extrems de preu de bitllet inferior.

```
titanic.bp<-boxplot(titanic$Age~titanic$Pclass, horizontal = TRUE,
                    col=c("green","orange","red"), main="Edat per classe",
                    xlab = "Edat", ylab = "Classe de Ticket")
```




```
titanic.bp$out
```

```
##  [1] 66.00  3.00  0.83  1.00  3.00 59.00  2.00  3.00  2.00 62.00 57.00 70.00
## [13] 60.00  0.67 57.00  1.00  0.83 59.00 70.50 55.50 65.00 61.00 63.00 74.00
```

En aquest gràfic podem observar que els valors extrems es troben a segona i a tercera classe. El valor dels outliers que es troben a segona classe tenen una edat superior que els valors que es troben a tercera classe, on els valors extrems es troben amb un inici d'edat inferior.

En aquest gràfic trobem també outliers a segona classe amb valors inferiors.

3.3 Dades perdudes - missing data.

Les dades contenen ceros o elements buits? Com gestionaries cadascun d'aquests casos?. Anem a evaluar quines variables tenen valors nulls o no informats (amb "") i les completarem i discretitzarem si escau. Ademés, de cara a fer prediccions i sobretot si volem fer regressió logística és adequat que les variables convertides **quedin com a tipus factor** en R.

```
#
#missing values
#
colSums(is.na(titanic))
```

```
## Survived   Pclass      Sex      Age     Fare Embarked    Titol  Isalone
##        0        0        0      177        0        0        0        0
```

```
#a train tenim bàsicament null values a Age, 177
colSums(titanic=="")
```

```
## Survived   Pclass      Sex      Age     Fare Embarked    Titol  Isalone
##        0        0        0       NA        0        2        0        0
```

```
#a train tenim valors perduts no nulls a Cabin i a Embarked
colSums(is.na(titanic_test))
```

```
## Survived   Pclass      Sex      Age     Fare Embarked    Titol  Isalone
##        0        0        0       86        1        0        0        0
```

```
#a train tenim bàsicament null values a Age 86 i 1 a Fare
colSums(titanic_test=="")
```

```
## Survived   Pclass      Sex      Age     Fare Embarked    Titol  Isalone
##        0        0        0       NA       NA        0        0        0
```

```
#a train tenim valors perduts no nulls a Cabin
#
# Embarked
#
#Al set de traing tenim 2 missing values que son "", no son pas NA
#Els completem amb el valor més freqüent
#Discretitzem Embarked - convertim Embarked a numeric (0-2 son 3 valors)
#Crearem una funció on passem una columna de dataframe i ens la retorna
#discretitzada
clean_Embarked <- function (cp) {
  #asignamos los 2 "" el valor més freqüent freq_embarked
  freq_embarked <- tail(names(sort(table(cp))), 1)
  cp[cp==""] <- freq_embarked
  #discretitzem
  cp[cp=='S'] <- 0
  cp[cp=='C'] <- 1
  cp[cp=='Q'] <- 2
  cp<-as.factor(cp)
  return(cp)
}
#Train
titanic %>% count(Embarked)
```

```
##   Embarked   n
## 1            2
## 2        0 644
```

```
## 3        1 168
## 4        2  77
```

```
titanic$Embarked <- clean_Embarked(titanic$Embarked)
titanic %>% count(Embarked)
```

```
##   Embarked   n
## 1        0 646
## 2        1 168
## 3        2  77
```

```
#Test
titanic_test %>% count(Embarked)
```

```
##   Embarked   n
## 1        0 270
## 2        1 102
## 3        2  46
```

```
titanic_test$Embarked <- clean_Embarked(titanic_test$Embarked)
titanic_test %>% count(Embarked)
```

```
##   Embarked   n
## 1        0 270
## 2        1 102
## 3        2  46
```

```
#
# Fare - crearem una nova variable però deixem l'original per a fer un test de
#hipòtesis posterior
#
#Al set de test tenim 1 NA
#Els completem amb el valor més freqüent
#Discretitzem Fare - creem una rang per Fare de 4 convertim Fare a factor
#(Q1-Q4 son 4 valors)
clean_Fare <- function (cp) {
  #com a paràmetre rep una columna d'un dataframe que serà l'afectada
  #asignamos els nulls o  "" el valor més freqüent freq_fare
  fare_embarked <- tail(names(sort(table(cp))), 1)
  #cp[cp==""] <- fare_embarked
  cp[is.na(cp)] <- fare_embarked
  #hem de convertit a numeric per poder fer els rangs
  cp <- as.numeric(cp)
  #discretitzem en base a generar un rang de 4 buckets bassat en quartiles
  #la funció CutQ genera els rangs i ja els assigna segons el valor
  fare_quartiles <- CutQ(cp)
  #és una variable tipus factor que assignem a la variable de sortida
  cp <- fare_quartiles
  return(cp)
}
#Train
titanic <- titanic %>%
  mutate(Fare_disc =  clean_Fare(titanic$Fare)
  )
#titanic$Fare <- clean_Fare(titanic$Fare)
titanic %>% count(Fare_disc)
```

```
##   Fare_disc   n
## 1        Q1 223
## 2        Q2 224
## 3        Q3 222
## 4        Q4 222
```

```
#Test
titanic_test <- titanic_test %>%
  mutate(Fare_disc =  clean_Fare(titanic_test$Fare)
  )
#titanic_test$Fare <- clean_Fare(titanic_test$Fare)
titanic_test %>% count(Fare_disc)
```

```
##   Fare_disc   n
## 1        Q1 115
## 2        Q2  96
## 3        Q3 102
## 4        Q4 105
```

Pels cas del missing values de la variable Age anem a fer una mica de tractament especial. Al tractar-se d'una variable cuantitativa continua ens interessa **omplir el missing values (que no son poc) d'una forma acurada** i per altre banda, pensant en els anàlisis posteriors **ens interessa discretitzar aquesta variable**. Per tant procedirem de la següent manera: 1. predicció de valors fent servir altres features correlades (Age, Sex, Pclass), agafarem per cada combinació de Pclass-Sex la mediana del valor de Age, i aquest serà el que assignarem per a totes les combinacinos de Pclass-Sex que tinguin missing values o NA 2. creem una nova feature AgeBand (5 intervals de edat) 3. substituim Age per AgeBand 4. finalment podem esborrar la AgeBand

```
#creem la función per al tractament de la variable Age
clean_Age <- function (df) {
  #com a paràmetre rebem un data-frame complet perque necessitarem varies columnes

  #Necessitem com a pas previ discretitzar la variable Sex
  #df$Sex[df$Sex=='male'] <- 0
  #df$Sex[df$Sex=='female'] <- 1
  #df$Sex<-as.integer(df$Sex)
  #1.matriu pels guessed values de age segons Sex i Pclass
  pred_age <- matrix(nrow = 3, ncol = 2)
  #2.càlcul de les medianes per a cada combinació de Pclass (i)
  for(i in 1:3) {
    for(j in 1:2) {
      df_ages <- subset(df,Pclass==i & Sex==j-1)
      pred_age[i,j]  <- median(df_ages$Age,na.rm=T)
      }
  }
  #3.canviem els NA per a cada combinació de Pclass i Sex pel valor que hem calculat abans
  for(i in 1:3) {
    for(j in 1:2) {
      df$Age[is.na(df$Age) & df$Pclass ==i & df$Sex == j-1] <- pred_age[i,j]
    }
  }

  #4.Creem els intervals de Age
  df$Age_grouping <- cut(df$Age, breaks=c(0,16,32,48,64,100,140), right = FALSE, labels = FALSE)
  #5.Assignem els intervals com a valor de Age
  df$Age <- df$Age_grouping
```

```
  #6.Esborrem la variable intermitja Age_grouping
  df = df[,!(names(df) %in% c("Age_grouping"))]

  return(df)
}
#Train
titanic <- clean_Age(titanic)
titanic %>% count(Age)
```

```
##   Age   n
## 1   1  83
## 2   2 492
## 3   3 227
## 4   4  76
## 5   5  13
```

```
head(titanic)
```

```
##   Survived Pclass Sex Age    Fare Embarked Titol Isalone Fare_disc
## 1        0      3   0   2  7.2500        0     1       1        Q1
## 2        1      1   1   3 71.2833        1     1       1        Q4
## 3        1      3   1   2  7.9250        0     2       0        Q2
## 4        1      1   1   3 53.1000        0     1       1        Q4
## 5        0      3   0   3  8.0500        0     1       0        Q2
## 6        0      3   0   2  8.4583        2     1       0        Q2
```

```
#Test
titanic_test <- clean_Age(titanic_test)
titanic_test %>% count(Age)
```

```
##   Age   n
## 1   1  32
## 2   2 251
## 3   3  91
## 4   4  39
## 5   5   5
```

```
head(titanic_test)
```

```
##   Survived Pclass Sex Age    Fare Embarked Titol Isalone Fare_disc
## 1        0      3   0   3  7.8292        2     1       0        Q1
## 2        1      3   1   3  7.0000        0     1       1        Q1
## 3        0      2   0   4  9.6875        2     1       0        Q2
## 4        0      3   0   2  8.6625        0     1       0        Q2
## 5        1      3   1   2 12.2875        0     1       1        Q2
## 6        0      3   0   1  9.2250        0     1       0        Q2
```

4. Anàlisi de les dades.

En aquest apartat realitzarem l'anàlisi o exploració de les dades, tot explicant les principals característiques d'aquestes dades ja netejades, per tal de poder resposndre a les preguntes plantejades d'aquesta pràctica.

Regresion logaritmica per a predir si sobreviu o no –> fer diversos models (age, sex-class-fare-age, tots i comparar el AIC u overall$accuracy de confusionMatrix

4.1 Selecció de dades.

Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar).

Els grups els tenim clarament identificats en el sentit de que farem servir el cojunt de train (titanic) per a generar els models, i el conjunt de test (titanic_test) per a provar-lo o ajustar-lo, i obtenir la seva bondat (accuracy).

Per centrant-nos en els diferents tipus d'anàlisis i en concret si volem fer un contrast de hipótesis per a validar si la mija dels preus (fare) dels que sobreviuen és més gran i igual que la dels que moren, crearem dos grups entorn a la variable preu : els preus dels que sobreviuen i els preus dels que moren

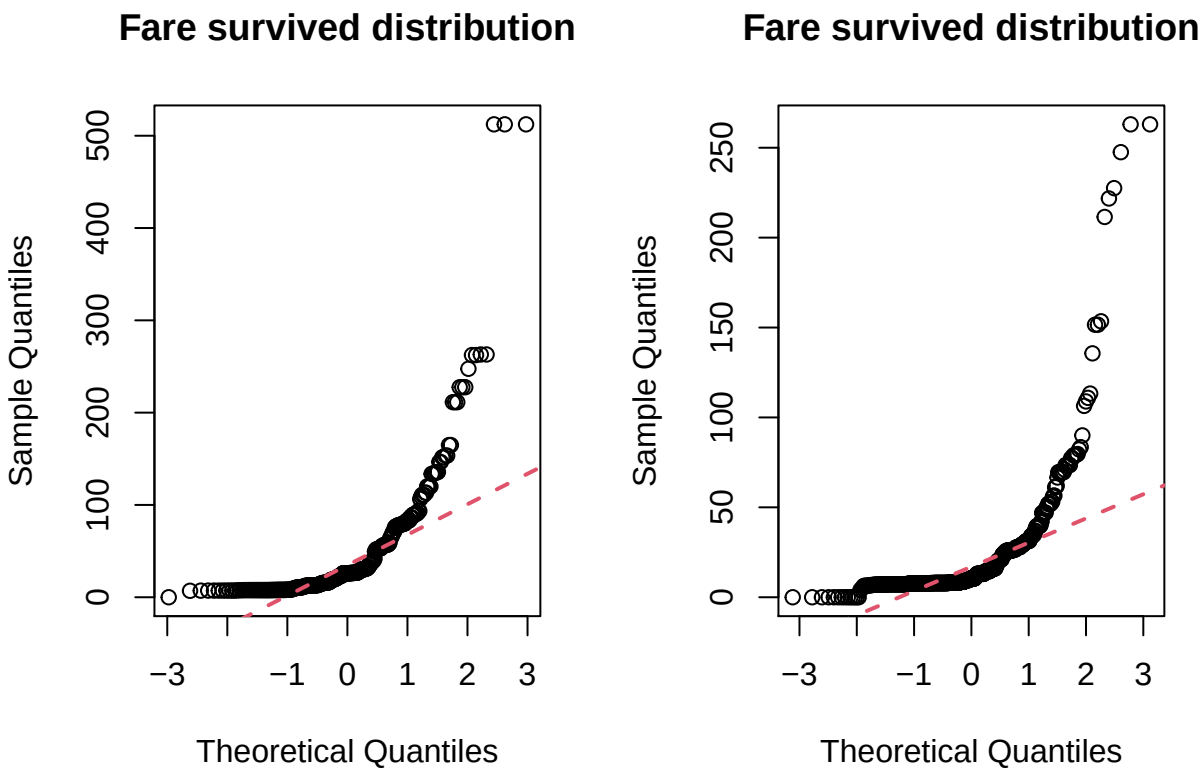
```
fare_vius = titanic$Fare[titanic$Survived==1]
fare_morts = titanic$Fare[titanic$Survived==0]
```

4.2 Comprovació de la normalitat i homegeneitat de la variança.

Farem inicialment aquestes comprovacions per la variable Fare que és la que voldrem involucrar en el contrast de hipòtesis.

Per la comprovació de normalitat ho farem de manera visual fent servir la funció qnorm

```
#1) Diagrama de punts de fare_vius i fare_morts
par(mfrow=c(1,2))
qqnorm(fare_vius,main="Fare survived distribution");qqline(fare_vius, col = 2,lwd=2,lty=2)
qqnorm(fare_morts,main="Fare survived distribution");qqline(fare_morts, col = 2,lwd=2,lty=2)
```



Encara que tenim força punts fora de la línea recta podriem dir que la majora s'agrupen al voltat d'ella per tant donarem per supossat el factor de normalitat, encara que amb dubtes. Amb un conjunt de dades gran podriem arrivar a assumir el factor de normalitat però amb només 891 observaciones aquesta afirmació queda en entredit. De totes formes farem el contrast supossant normalitat. Per la comprovació de la homocedasticitat podem fer servir la funció var.test de R.

```
#Comprovem homocedasticitat - variances iguals
var.test(x = fare_vius,y = fare_morts)
```

```
##
##  F test to compare two variances
##
## data:  fare_vius and fare_morts
## F = 4.5017, num df = 341, denom df = 548, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  3.725366 5.463382
## sample estimates:
## ratio of variances
##           4.501697
```

De la sortida de var.test podem veure que **el ratio of variances que és de 4.50 està dintre del interval de confiança de 95%**, per tant el test no troba diferencies significatives entre les variances d'ambdos grups.

4.3 Aplicació proves estadístiques.

Aplicació de proves estadístiques per a comparar els grups de dades. En funció de les dades i l'objectiu de l'estudi, aplicar proves de contrast de hipòtesis, correlacions, regressions, etc. Aplicar almenys 3 mètodes

d'anàlisis diferents.

4.3.1 Contrastos de hipotesis.

Podem acceptar que el preu del passatge (Fare) és més gran en els que sobreviuen que en els que moren? Validarem aquest fet amb un contrast de hipòtesis per a la diferència de 2 mitjanes amb els suposits de normalitat i homocedasticitat, és a dir, aplicarem una t de Student. La hipòtesi nul.la H0 seria establir que **la mitjana de preus del passatge dels que sobreviuen es igual a la dels que no sobreviuen.** H0 : mean(Fare sobreviuen) = mean(Fare moren)

ó

H0 : mean(Fare sobreviuen) - mean(Fare moren) = 0 La hipòtesi alternativa H1 representa que s'ha produit algun canvi respecte la situació descrita per la hipòtesis nul.la. En aquest cas establirem el fet que realment volem comprovar al final, que la mitjana del preu del passatge del que sobreviuen és més gran que la dels que moren (és un contrast unilateral per la dreta). H1 : mean(Fare sobreviuen) > mean(Fare moren)

ó

H1 : mean(Fare sobreviuen) - mean(Fare moren) > 0 Encara que podem utilitzar directament la funció t.test de R ens farem una funció propia que ens implementi el càlcul del nostre contrast

```
#Contruim una funció que ens faci el contrast de la diferencia de
#mitjanes de 2 mostres en mode unilateral dret
contrast_dif_mitjana_2_mostres <- function (p_mostra1,p_mostra2, p_alfa,p_tipus)
  {

  #p_mostra1 és la mostra 1
  #p_mostra2 és la mostra 2
  #p_alfa és el nivell de significació

  #H0 : mean(p_mostra1) - mean(p_mostra2) = 0 - hipòtesis nul.la
  #H1 : mean(p_mostra1) - mean(p_mostra2) > 0 - hipòtesis alternativa
  #Calculem tamany,mitjanes i desviacions típiques d'ambues mostres
  n_us = length(p_mostra1)
  n_nous = length(p_mostra2)
  mean_us = mean(p_mostra1)
  mean_nous = mean(p_mostra2)
  dev_tipica_us = sqrt(sum((p_mostra1-mean_us)^2)/(n_us-1))
  dev_tipica_nous = sqrt(sum((p_mostra2-mean_nous)^2)/(n_nous-1))
  #calculem estadístic t
  #ditribució t-student amb n_us+n_nous - 2 graus de llibertat (398)
  s = sqrt(((n_us-1)*dev_tipica_us^2+((n_nous-1)*dev_tipica_nous^2))/(n_us+n_nous-2))
  s_error_std = s * sqrt((1/n_us)+(1/n_nous))

  t = (mean_us - mean_nous) / s_error_std

  #i finalment el p-value tenint en compte la distribució de t
  #i la hipotesis alternativa
  p_valor = case_when (p_tipus == 'uniesquerra' ~ pt(t,n_us+n_nous-2),
                       p_tipus == 'unidreta' ~ pt(-t,n_us+n_nous-2),
                       p_tipus == 'bidireccional' ~ 2*pt(t,n_us+n_nous-2)
                       )

  #si p_value >= nivell de significació p_alfa, acceptarem la H0
```

```
  #si p_value < nivell de significació p_alfa, rebutjarem la H0


  if (p_valor >= p_alfa) {
    ic = c(t,p_valor,'Acceptem Hipòtesi nul.la')
  }
  else
  {
    ic = c(t,p_valor,'Rebutgem Hipòtesi nul.la')
  }

  return(ic)

}
c_mitjanes = contrast_dif_mitjana_2_mostres(fare_vius,fare_morts,0.5,'unidreta')
c_mitjanes
```

```
## [1] "7.93919166087105"         "3.06009467096209e-15"
## [3] "Rebutgem Hipòtesi nul.la"
```

```
#[1] t(estadístic) = "7.93919166087105"         p_valor = "3.06009467096209e-15"
#[3] Resultat : "Rebutgem Hipòtesi nul.la"
#el p_value es pot dir que és infim i per tant més petit que 0.05 així que rebutjariem
#la Hipotesis nul.la
#Comprovació amb t_test
t.test( fare_vius, fare_morts, # dues mostres
        alternative = "greater", # contraste per resta de mitjanes
        paired = FALSE, # muestras independientes
        var.equal = TRUE,  # se supone homocedasticidad
        conf.level=0.95)
```

```
##
##  Two Sample t-test
##
## data:  fare_vius and fare_morts
## t = 7.9392, df = 889, p-value = 3.06e-15
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  20.82763      Inf
## sample estimates:
## mean of x mean of y
##  48.39541  22.11789
```

4.3.1.1 Interpretar contratos de hipotesis. Com el **p-value (3.06e-15, que és la probabilitat del resultat del estadístic t quan la hipòtesis nul.la és certa) és més PETIT que el nivell d'acceptació (0,05) llavors REBUTGEM la Hipòtesis nul.la** per tant això vol dir que confirmem que els preus del passatges del que sobreviuen és més gran el preu dels que moren. Per una altre banda, la sortida de la funció t.test ens està dient que **el p_value NO està dintre de l'interval d'acceptació de la hipòtesi nul.la**, per tant ens porta a rebutjar-la. Així mateix, el fet de que **l'estadístic de contrast sigui gran (7.9392)** fa que estigui allunyat del cero (zona on la distribució normal estàndard concentra una probabilitat més gran), per tant **poc probable sota la hipòtesis nul.la.** El fet d'haver plantejar una hipòtesi alternativa unilateral per la dreta fa també que aquest fet de tenir un valor positiu per l'estadístic de contrast, aquest sigui més probable sota la alternativa.

4.3.2 Correlacions.

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:DescTools':
##
##     %nin%, Label, Mean, Quantile
```

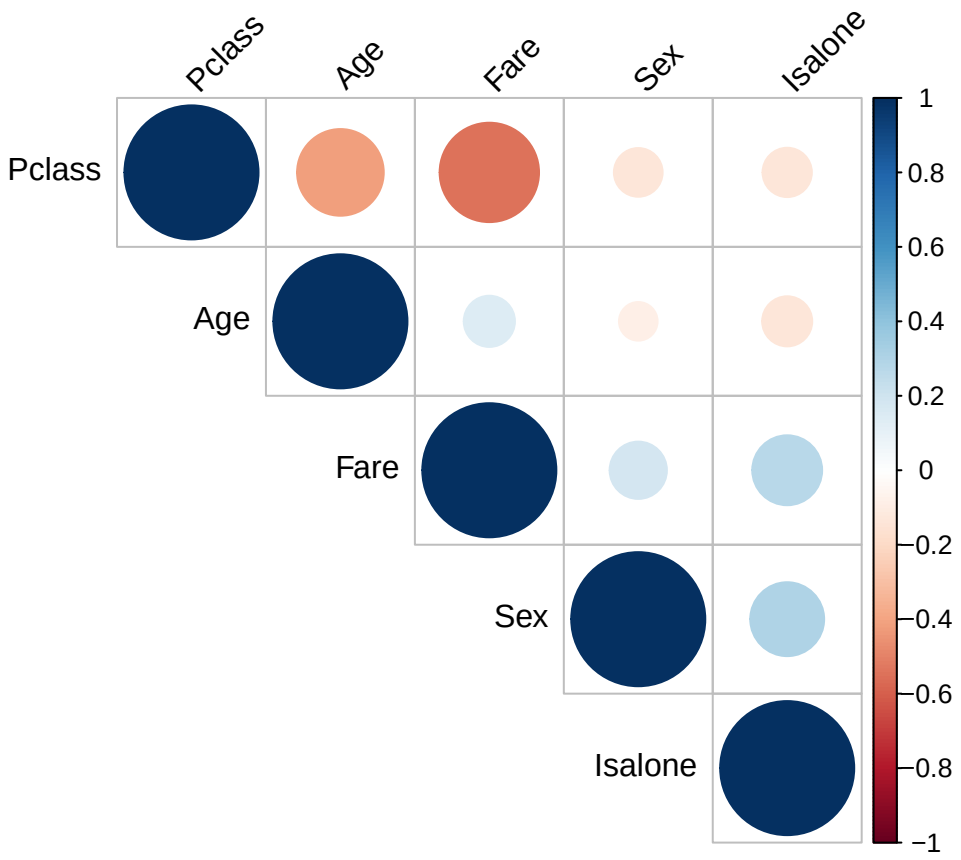
```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
#creem un nou dataframe amb les columnes numèriques
titanic_cor <- titanic[, c("Pclass", "Sex", "Age","Fare","Isalone")]
rcor <- cor(titanic_cor)
rcor
```

```
##             Pclass         Sex         Age       Fare    Isalone
## Pclass   1.0000000 -0.13190049 -0.41790461 -0.5494996 -0.1352072
## Sex     -0.1319005  1.00000000 -0.08284438  0.1823328  0.3036462
## Age     -0.4179046 -0.08284438  1.00000000  0.1458387 -0.1394354
## Fare    -0.5494996  0.18233283  0.14583865  1.0000000  0.2718324
## Isalone -0.1352072  0.30364619 -0.13943544  0.2718324  1.0000000
```

```
#i en format gràfic
corrplot(rcor, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



4.3.2.1 Interpretar correlacions. En aquest gràfic buscarem els cercles que siguin més grans i els de tonalitats més brillants, i ens fixem en el següent :

- Fare i Pclass estan fortament relacionades. Les cabines de primera classe seran les que tinguin una tarifa més cara respecte a les de les altres classes.
- Pclass i Age estan també relacionades. La gent més adinerada tindrà una franja d'edat més elevada.
- Sex i Isalone tindrien una relació molt baixa. La gent sigui dona o home no té relació en classificar si està sola sense familiars a bord. Igual passarà amb Fare i Isalone, on no hi haurà una correlació.

4.3.3 Regressió Logarítmica.

```
#Necessitem convertir primer Survive com a factor
titanic$Survived <- as.factor(titanic$Survived)
titanic_test$Survived <- as.factor(titanic_test$Survived)

titanic.train <- glm(Survived ~ Pclass + Sex + Age, family = binomial, data = titanic)
summary(titanic.train)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age, family = binomial,
##     data = titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.5742  -0.6340  -0.4574   0.6105   2.3512
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.3737     0.4923   4.822 1.42e-06 ***
## Pclass       -1.1967     0.1245  -9.614  < 2e-16 ***
## Sex           2.5935     0.1857  13.964  < 2e-16 ***
## Age          -0.4942     0.1224  -4.037 5.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  809.96  on 887  degrees of freedom
## AIC: 817.96
##
## Number of Fisher Scoring iterations: 4
```

```
#odds ratio - per a interpretar el model
exp(cbind("Odds ratio" = coef(titanic.train), confint.default(titanic.train, level = 0.95)))
```

```
##             Odds ratio     2.5 %     97.5 %
## (Intercept) 10.7369785 4.0911539 28.1785309
## Pclass       0.3021996 0.2367817  0.3856912
## Sex         13.3760404 9.2948427 19.2492185
## Age          0.6100329 0.4798972  0.7754581
```

```
#Importancia de les variables
varImp(titanic.train)
```

```
##          Overall
## Pclass  9.614397
## Sex    13.964349
## Age     4.037230
```

```
#Confusion matrix i accuracy del model
confusionMatrix(as.factor(as.numeric(predict(titanic.train, titanic_test)>=0.5)), titanic_test$Survived
```

```
##  Accuracy
## 0.8397129
```

```
#0.84 --> molt bona accuracy per aquest model

#Exemples de prediccions

#predir la probabilitat de sobreviure de una dona de classe 3 de 32 anys
pred_sobr1<-predict(titanic.train, data.frame(Pclass=3,Sex=1,Age=2),type ="response")
pred_sobr1
```

```
##         1
## 0.5959628
```

```
#0.59

#predir la probabilitat de sobreviure d'un home de classe 1 de 52 anys
pred_sobr2<-predict(titanic.train, data.frame(Pclass=1,Sex=0,Age=4),type ="response")
pred_sobr2
```

```
##         1
## 0.3100376
```

```
#0.31
```

4.3.1.3. Interpretar regressió logaritmica Normalment encara que podem extreure la Accuracy d'un model mitjançant la matriu de confussió (que en aquest cas resulta un model força ajustat amb un 0.84 d'accuracy), quan parlem d'una regressió logarítmica és habitual fer la interpretació mirant el ods-rati que hem obtingut.

Els *odds son la raó de la probabilitat d'ocurrència d'un succés entre la probabilitat de la seva NO ocurrència.*

Els **odds-rati (OR) es calculen com la raó entre els odds**, on la variable de resposta (la depenent) Y està present entre els individus i la variable independent X (els regressors) pot estar present o no, és a dir, prendre els valors X=1 e X=0.

Això ens dona un criteri de interpretació d'aquests odds-rati: 1. Un OR = 1 significa que **NO hi ha associación entre la variable resposta i la covaraible**. 2. Un **OR inferior a la unitat s'interpreta com un factor de protecció**, és a dir, el succés és menys probable en presència d'aquesta covariable. 3. Un **OR major a la unitat s'interpreta com un factor de risc**, és a dir, el succés és més probable en presència d'aquesta covariable.

Segons això pel nostre model tindriem:

1. **Pclass**: el seu odds ratio és de 0.30, això vol dir que és menys probable que el passatge sobrevisqui en presència d'aquest regressor. De fet es pot concloure que **per cada unitat que augmenta el Pclass (la classe on viatja)**, si les altres variables es mantenen constants, **l'odds de sobreviure és 0.30 vegades menor**.

Com el IC (Interval de Confiança) per aquest OR és de (0.23-0.38) és pot dir que **si augmenta el Pclass disminueix la probabiliat de sobreviure entre 0.23 i 0.38 vegades.**

2. **Age** : el seu odds ratio és de 0.61, això vol dir que és menys probable que el passatger sobrevisqui en presència d'aquest regressor. De fet es pot concloure que **per cada unitat que augmenta la Age (el interval de edat)**, si les altres variables es mantenen constants, **l'odds de sobreviure és 0.61 vegades menor**.

Com el IC (Interval de Confiança) per aquest OR és de (0.47-0.77) és pot dir que **si augmenta el rang de l'edadt disminueix la probabiliat de sobreviure entre 0.47 i 0.77 vegades.**

3. **Sex** : el seu odds ratio és de 13.37, és a dir més gran que 1 amb diferència, això vol dir que és més probable que el passatger sobrevisqui en presència d'aquest regressor. De fet es pot concloure que **per cada unitat que augmenta el Sex (o sigui que sigui una dona)**, si les altres variables es mantenen constants, **l'odds de sobreviure és 13.37 vegades més gran**. Aquest indicador ens està dient que el fet de ser dona te una influència important en el fet de sobreviure.

Com el IC (Interval de Confiança) per aquest OR és de (9.29-19.24) és pot dir que **si augmenta el Sex (o sigui que sigui una dona) augmenta la probabiliat de sobreviure entre 9.29 i 19.24 vegades.**

De la sortida de la funció varImp també es conclou que és la variable que més efecte té en el model.

En la simulació de prediccions es pot veure la diferència de probabilitats quan la observació és un home i una dona. **En el cas d'home tenim una prob. de 0.31 i en cas de dona 0.59, quasi el doble**.

Anem a afegir totes les variables al model per veure si tenim canvis

```
titanic_lm_full <- glm(Survived ~ ., family = binomial, data = titanic)
summary(titanic_lm_full)
```

4.3.1.4. Model logaritmic amb totes les variables

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4437  -0.6324  -0.3627   0.5844   2.5565
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.921196   0.699731   2.746  0.00604 **
## Pclass       -1.306229   0.175736  -7.433 1.06e-13 ***
## Sex           3.559889   0.322070  11.053  < 2e-16 ***
## Age          -0.303818   0.142136  -2.138  0.03256 *
## Fare          0.001880   0.002514   0.748  0.45457
## Embarked1     0.617735   0.241615   2.557  0.01057 *
## Embarked2     0.452108   0.344770   1.311  0.18975
## Titol2       -0.751302   0.333994  -2.249  0.02448 *
## Titol3        9.677179 535.411399   0.018  0.98558
## Titol4        2.749845   0.451051   6.097 1.08e-09 ***
## Titol5       -0.309076   0.493431  -0.626  0.53107
## Isalone      -0.491830   0.275869  -1.783  0.07461 .
## Fare_discQ2   0.093354   0.303961   0.307  0.75875
## Fare_discQ3  -0.125765   0.352646  -0.357  0.72137
## Fare_discQ4  -0.426367   0.445503  -0.957  0.33854
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  757.11  on 876  degrees of freedom
## AIC: 787.11
##
## Number of Fisher Scoring iterations: 12
```

```
#odds ratio - per a interpretar el model
exp(cbind("Odds ratio" = coef(titanic_lm_full), confint.default(titanic_lm_full, level = 0.95)))
```

```
##               Odds ratio      2.5 %     97.5 %
## (Intercept) 6.829120e+00  1.7328173 26.9139036
## Pclass      2.708395e-01  0.1919220  0.3822074
## Sex         3.515930e+01 18.7022421 66.0977453
## Age         7.379950e-01  0.5585560  0.9750797
## Fare        1.001882e+00  0.9969569  1.0068314
## Embarked1   1.854722e+00  1.1550906  2.9781150
## Embarked2   1.571622e+00  0.7996114  3.0889946
## Titol2      4.717520e-01  0.2451420  0.9078409
## Titol3      1.594944e+04  0.0000000        Inf
## Titol4      1.564020e+01  6.4611122 37.8597225
## Titol5      7.341251e-01  0.2791004  1.9309880
## Isalone     6.115061e-01  0.3561073  1.0500760
## Fare_discQ2 1.097850e+00  0.6050772  1.9919353
```

```
## Fare_discQ3 8.818224e-01  0.4417822  1.7601675
## Fare_discQ4 6.528768e-01  0.2726583  1.5633051
```

```
#Importancia de les variables
varImp(titanic_lm_full)
```

```
##                Overall
## Pclass       7.43288552
## Sex         11.05315294
## Age          2.13751192
## Fare         0.74781421
## Embarked1    2.55669426
## Embarked2    1.31133110
## Titol2       2.24944894
## Titol3       0.01807429
## Titol4       6.09653081
## Titol5       0.62638072
## Isalone      1.78284154
## Fare_discQ2  0.30712399
## Fare_discQ3  0.35663137
## Fare_discQ4  0.95704672
```

```
#Confusion matrix i accuracy del modelo
confusionMatrix(as.factor(as.numeric(predict(titanic_lm_full, titanic_test)>=0.5)), titanic_test$Survive
```

```
##  Accuracy
## 0.8896882
```

```
#0.8896 --> molt bona accuracy per aquest model

#Exemples de prediccions

#predir la probabilitat de sobreviure de una dona de classe 3 de 32 anys
pred_sobr1_f<-predict(titanic_lm_full, data.frame(Pclass=3,Sex=1,Age=2,Fare=7.90,Embarked=as.factor("2"
pred_sobr1_f
```

```
##         1
## 0.7170536
```

```
#0.99 amb titol 3, 0.80 amb titol 1, 0.71 amb isalone = 1

#predir la probabilitat de sobreviure d'un home de classe 1 de 52 anys
pred_sobr2_f<-predict(titanic_lm_full, data.frame(Pclass=1,Sex=0,Age=4,Fare=71.90,Embarked=as.factor("2
pred_sobr2_f
```

```
##         1
## 0.2826801
```

```
#0.28
```

Continuem tenim la variable Sex com la que més afecta en el model però a diferencia del model inicial ara la diferència amb d'altre variable com Pclass i els títols no és tan gran.

Hem millorat lleugerament l'accuracy, passant ara a 0.88, el fet d'afegir més variable fa baixar els graus de llibertat i per tant millora el model.

En les simulacions de prediccions observem la gran influencia del Sex al resultat, a més augmentat l'efecte per les variables Titol e IsAlone. Per exemple **per a una dona amb titol 3 i que NO viatja sola la probabilitat de sobreviure es de 0.99**, que baixa a 0.80 si tingués titol 1 i a 0.71 si viatgés sola.

A continuació probarem un parell més de models per a comprovar si obtenim algun de més accuracy que la regressió logística.

4.3.4 Random Forest.

```
library(caret)

set.seed(14, sample.kind = "Rounding")
```

```
## Warning in set.seed(14, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
gbmGrid <- expand.grid(
  mtry = c(1:7)
  )

train_rf <- train(Survived ~ ., method = "rf",
                      data = titanic,
                      tuneGrid = gbmGrid,
                      ntree = 100
)

train_rf$results
```

```
##   mtry  Accuracy     Kappa AccuracySD    KappaSD
## 1    1 0.7390946 0.3943908 0.02948303 0.06482983
## 2    2 0.8041305 0.5710542 0.01688670 0.04111380
## 3    3 0.8152310 0.5987050 0.01819009 0.04176063
## 4    4 0.8201350 0.6111018 0.01548529 0.03512870
## 5    5 0.8224079 0.6172157 0.01593606 0.03489820
## 6    6 0.8211576 0.6154690 0.01665366 0.03767441
## 7    7 0.8190142 0.6126313 0.01482194 0.03416299
```

```
train_rf$bestTune
```

```
##   mtry
## 5    5
```

```
#mtry =5 giving 0.8224079 accuracy

# compute accuracy on test_set

pr <- predict(train_rf, titanic_test)
mean(pr == titanic_test$Survived)
```

```
## Warning in `==.default`(pr, titanic_test$Survived): longitud de objeto mayor no
## es múltiplo de la longitud de uno menor
```

```
## Warning in is.na(e1) | is.na(e2): longitud de objeto mayor no es múltiplo de la
## longitud de uno menor
```

```
## [1] 0.6411483
```

```
#0.64, molt més baixa que amb regressió logística

#varImp
varImp(train_rf)
```

```
## rf variable importance
##
##             Overall
## Sex         100.000
## Fare         74.182
## Pclass       40.306
## Age          19.726
## Titol2       16.208
## Isalone       8.676
## Titol4        8.178
## Embarked0     6.512
## Fare_discQ4   5.227
## Embarked1     5.177
## Fare_discQ3   4.449
## Fare_discQ2   4.092
## Embarked2     3.985
## Titol5        1.965
## Titol3        0.000
```

No obtenim millor accuracy i veiem que en aquest model Sex i Fare son les variables amb més rellevencia i amb certa diferència.

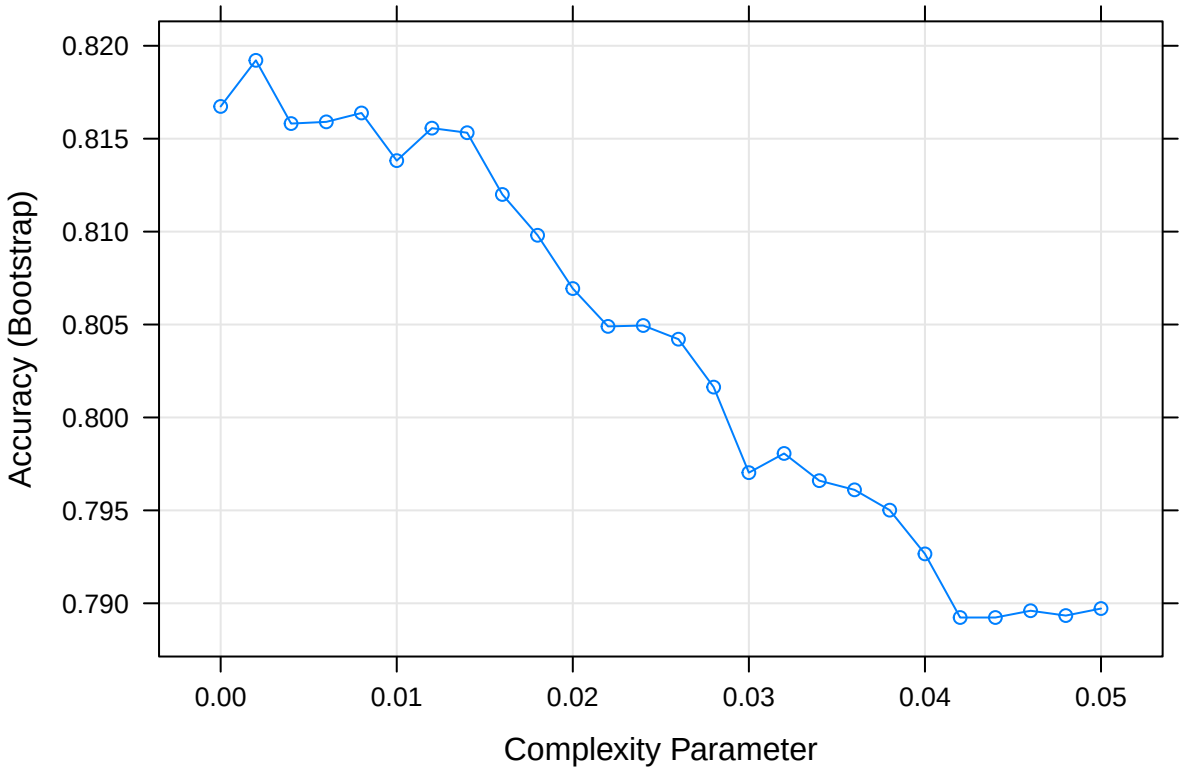
La raó de que tinguin pitjor accuracy amb les dades de test pot ser degut a que hem tingut overfitting. En aquest cas tindrem un random forest massa profund, s'assemblarà molt al conjunt de training però donarà prediccions dolentes amb les dades de test.

4.3.5 Classification Tree.

```
set.seed(10, sample.kind = "Rounding")
```

```
## Warning in set.seed(10, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
train_rpart <- train(Survived ~ .,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)),
                     data = titanic)
plot(train_rpart)
```




```
train_rpart$bestTune
```

```
##      cp
## 2 0.002
```

```
#predictors
predictors(train_rpart$finalModel)
```

```
##  [1] "Sex"         "Titol4"      "Fare"        "Pclass"      "Embarked0"
##  [6] "Titol2"      "Isalone"     "Age"         "Fare_discQ4" "Embarked1"
## [11] "Fare_discQ3" "Fare_discQ2" "Embarked2"
```

```
pr_cl <- predict(train_rpart, titanic_test)
mean(pr_cl == titanic_test$Survived)
```

```
## Warning in `==.default`(pr_cl, titanic_test$Survived): longitud de objeto mayor
## no es múltiplo de la longitud de uno menor
```

```
## Warning in is.na(e1) | is.na(e2): longitud de objeto mayor no es múltiplo de la
## longitud de uno menor
```

```
## [1] 0.6339713
```

```
# compute accuracy on test_set
# 0.6339
```

5. Representació dels resultats a partir de taules i gràfiques.

5.1. Matriu de confusió del model de regressió logística.

```
Prediccio <- predict(titanic.train, newdata = titanic, type = "response")

# establim el punt de tall
test_threshold <- 0.5
titanic.pred <- ifelse(Prediccio>test_threshold, 1,0)

# creem la matriu de confusió
matriuconf <- confusionMatrix(data=factor(titanic.pred), reference = factor(titanic$Survived))
matriuconf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 466 100
##          1  83 242
##
##                Accuracy : 0.7946
##                  95% CI : (0.7666, 0.8207)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5617
##
##  Mcnemar's Test P-Value : 0.2369
##
##             Sensitivity : 0.8488
##             Specificity : 0.7076
##          Pos Pred Value : 0.8233
##          Neg Pred Value : 0.7446
##              Prevalence : 0.6162
##          Detection Rate : 0.5230
##    Detection Prevalence : 0.6352
##       Balanced Accuracy : 0.7782
##
##        'Positive' Class : 0
##
```

El model té un 79,46% d'exactitud a la correcta classificació de les dades. Es va predir de manera correcta que 466 persones van sobreviure La taxa de sensibilitat a la classificació de les dades del 84,88%, classificant les dades de manera altament correcte.

5.2. Representació del model de classificació.

Dibuixem el arbre de decissió

```
predictors(train_rpart$finalModel)
```

```
##  [1] "Sex"         "Titol4"      "Fare"        "Pclass"      "Embarked0"
##  [6] "Titol2"      "Isalone"     "Age"         "Fare_discQ4" "Embarked1"
## [11] "Fare_discQ3" "Fare_discQ2" "Embarked2"
```

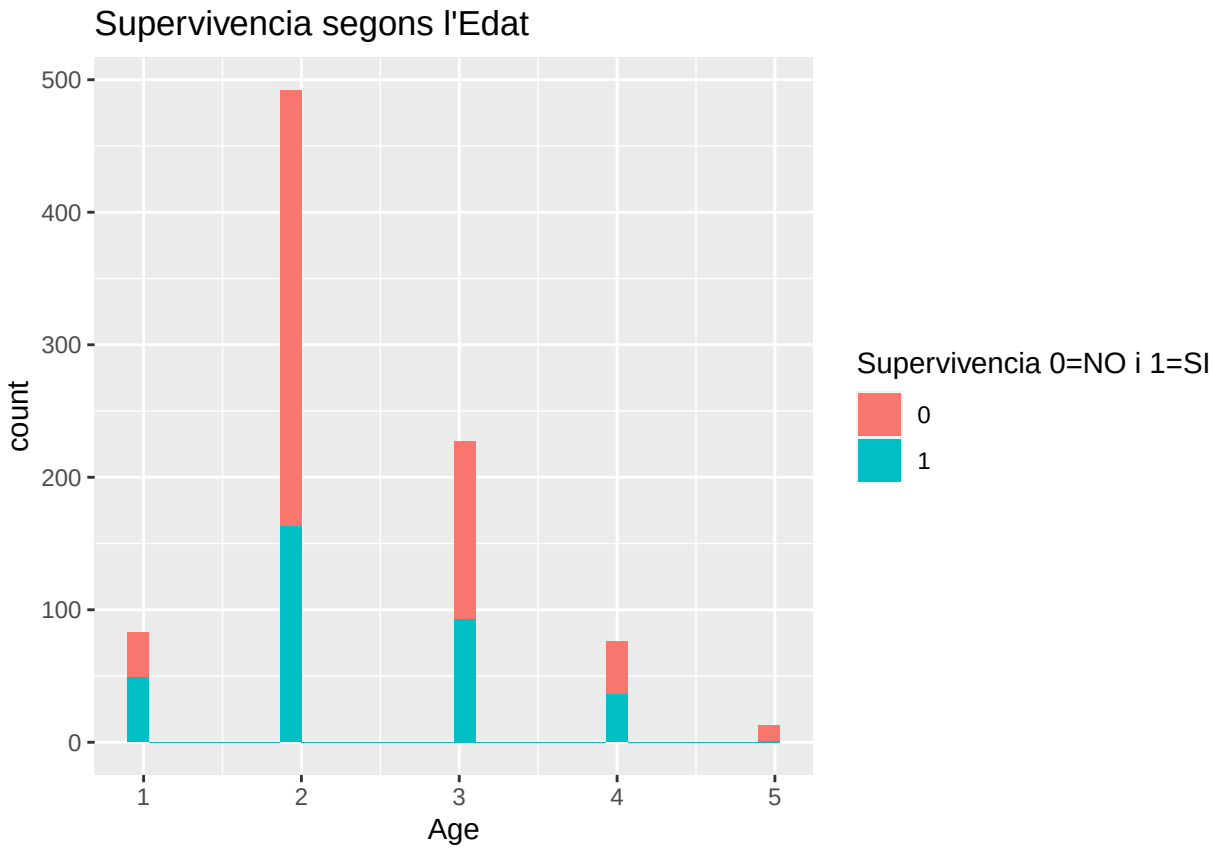
```r
plot(train_rpart$finalModel, margin = 0.1)
text(train_rpart$finalModel)
```

5.3. Gràfiques.

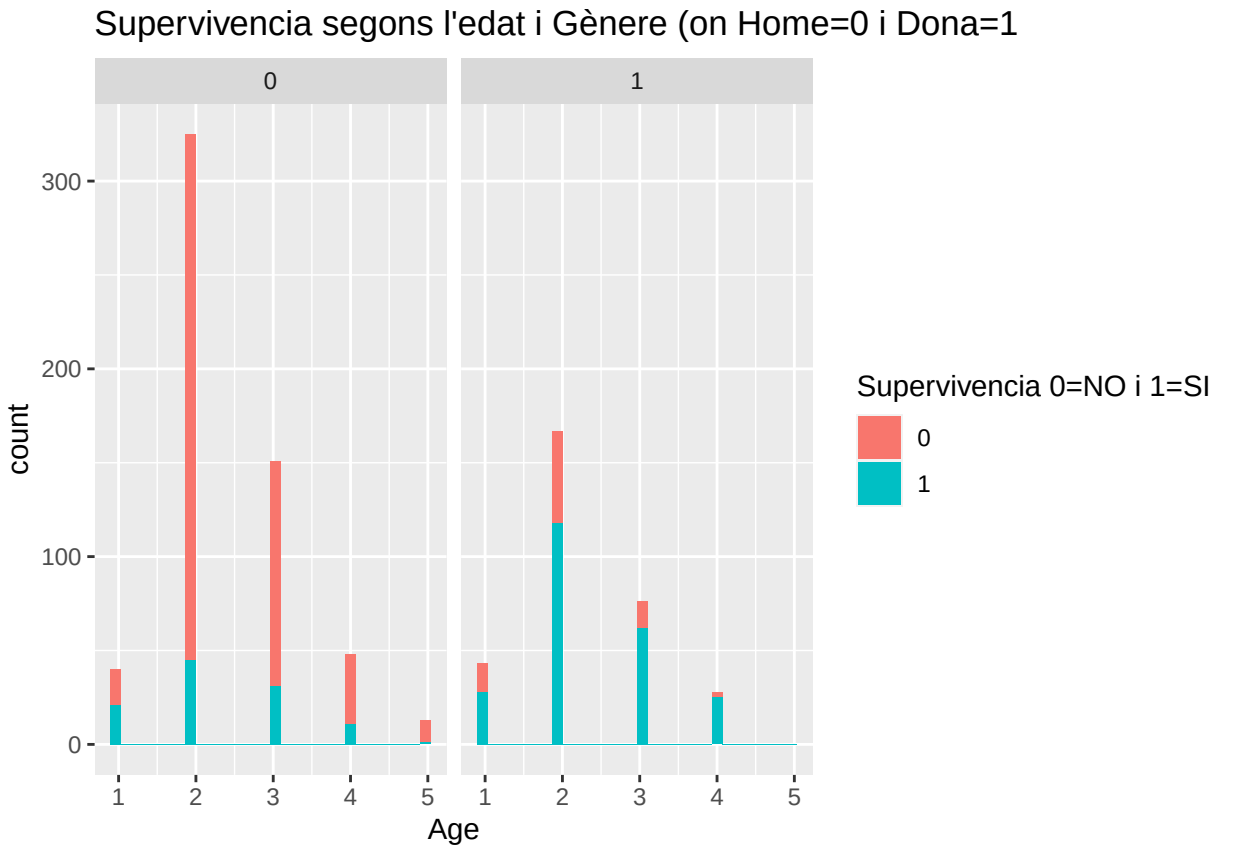
En aquest cas podem fer diverses gràfiques entre la variable depenent Survived i altres variables com per exemple:

- Supervivencia segons edat.
- Supervivencia segons la tarifa del passatge.
- Supervivencia segons la Classe del passatger.
- Supervivencia segons el Port d'embarcament del passatger.
- Supervivencia segons el titol de les persones i la classe on es trobaven allotjats al vaixell.

```r
library(ggplot2)

# supervivencia per edat i també per gènere

ggplot(titanic, aes(x=Age, fill=factor(Survived))) +
  geom_histogram(bins=30)+
  ggtitle("Supervivencia segons l'Edat")+
  scale_fill_discrete(name="Supervivencia 0=NO i 1=SI")
```




```
ggplot(titanic, aes(x=Age, fill=factor(Survived))) +
  geom_histogram(bins=30)+
  facet_grid(.~Sex)+
  ggtitle("Supervivencia segons l'edat i Gènere (on Home=0 i Dona=1")+
  scale_fill_discrete(name="Supervivencia 0=NO i 1=SI")
```




```
t.test(Age ~ Survived, data=titanic)
```

```
##
##  Welch Two Sample t-test
##
## data:  Age by Survived
## t = 0.78697, df = 682.79, p-value = 0.4316
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.06800435  0.15898317
## sample estimates:
## mean in group 0 mean in group 1
##        2.393443        2.347953
```

Realitzem dues gràfiques, una per veure la supervivencia segons l'edat del passatger, i la segona per veure la supervivència segons l'edat del passatger i també del gènere (si eren homes o dones).

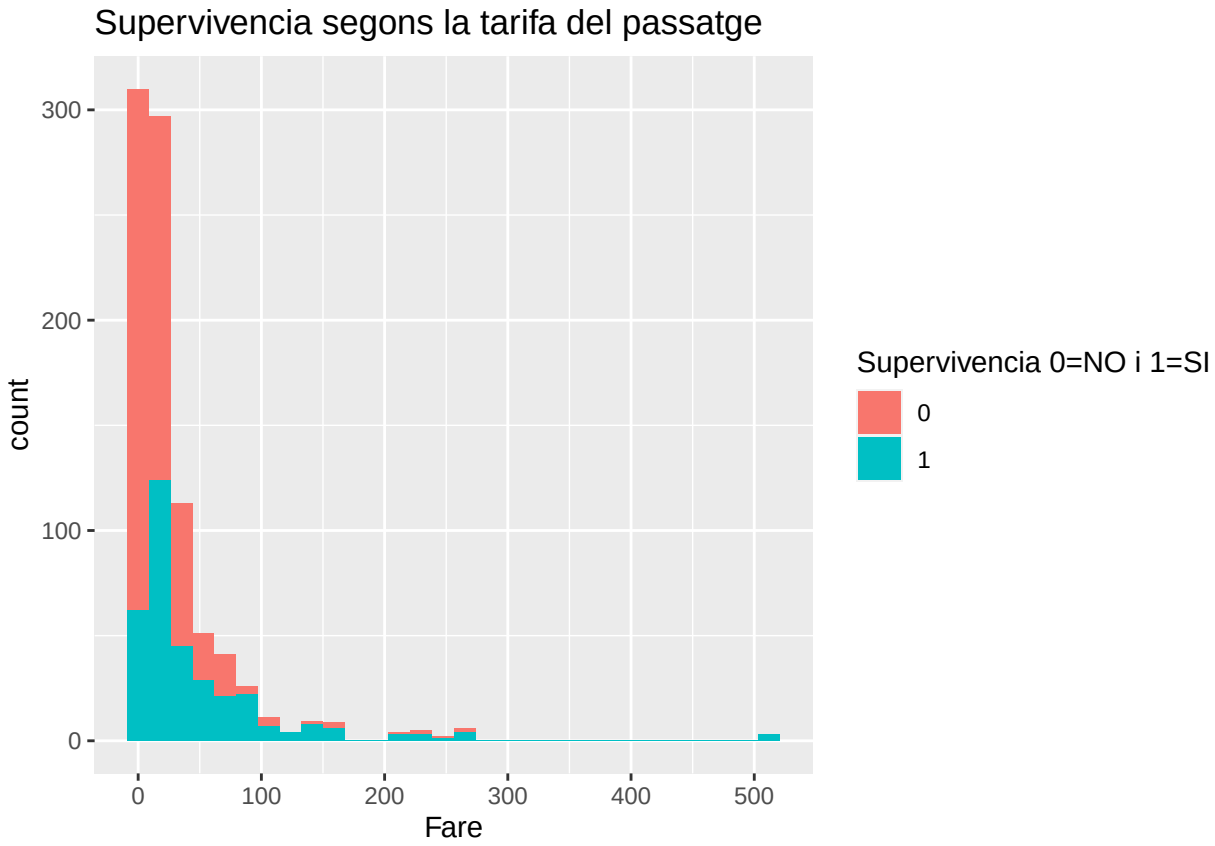
Veient els resultats del t-test on ens indica que el p-value és 0,04119, inferior a 0,5, es pot treure la conclusió que l'edat dels passatgers supervivents era inferior, i per tant, els supervivents eren més joves que els passatgers que van morir.

Aquestes dades es poden comprovar també a la gràfica "Supervivencia segons l'Edat".

En el gràfic amb títol "Supervivencia segons l'edat i Gènere", veiem dos gràfics de barres en paral · lel, on el gràfic de l'esquerre correpon als càlculs en homes (supervivencia per edat) i en el gràfic de la dreta es veuen el resultats de les dades per dones supervivents segons l'edat.

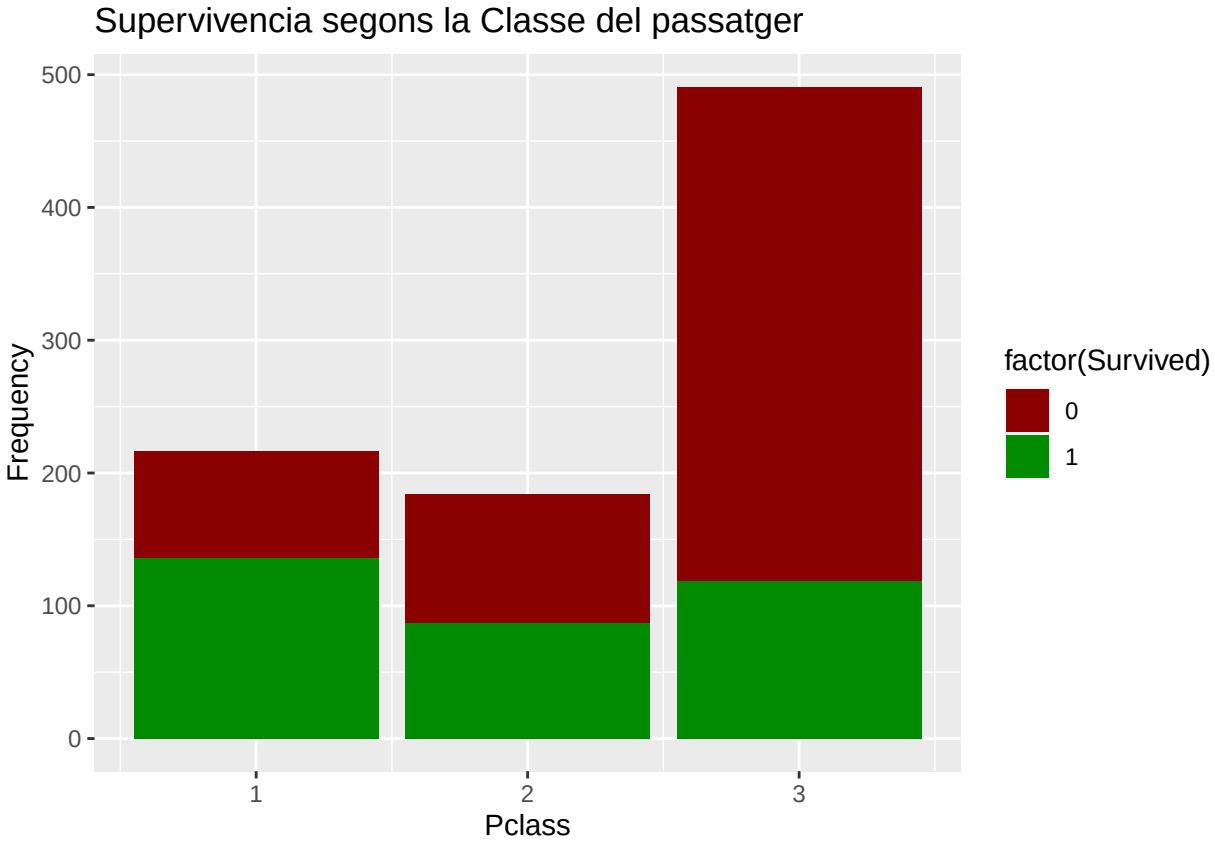
Podem concloure que la supervivencia de les dones (1) és major que la dels homes (0).

```r
# supervivencia per Tarifa del passatge
ggplot(titanic, aes(x=Fare, fill=factor(Survived)))+
  geom_histogram(bins=30)+
  ggtitle("Supervivencia segons la tarifa del passatge")+
  scale_fill_discrete(name="Supervivencia 0=NO i 1=SI")
```



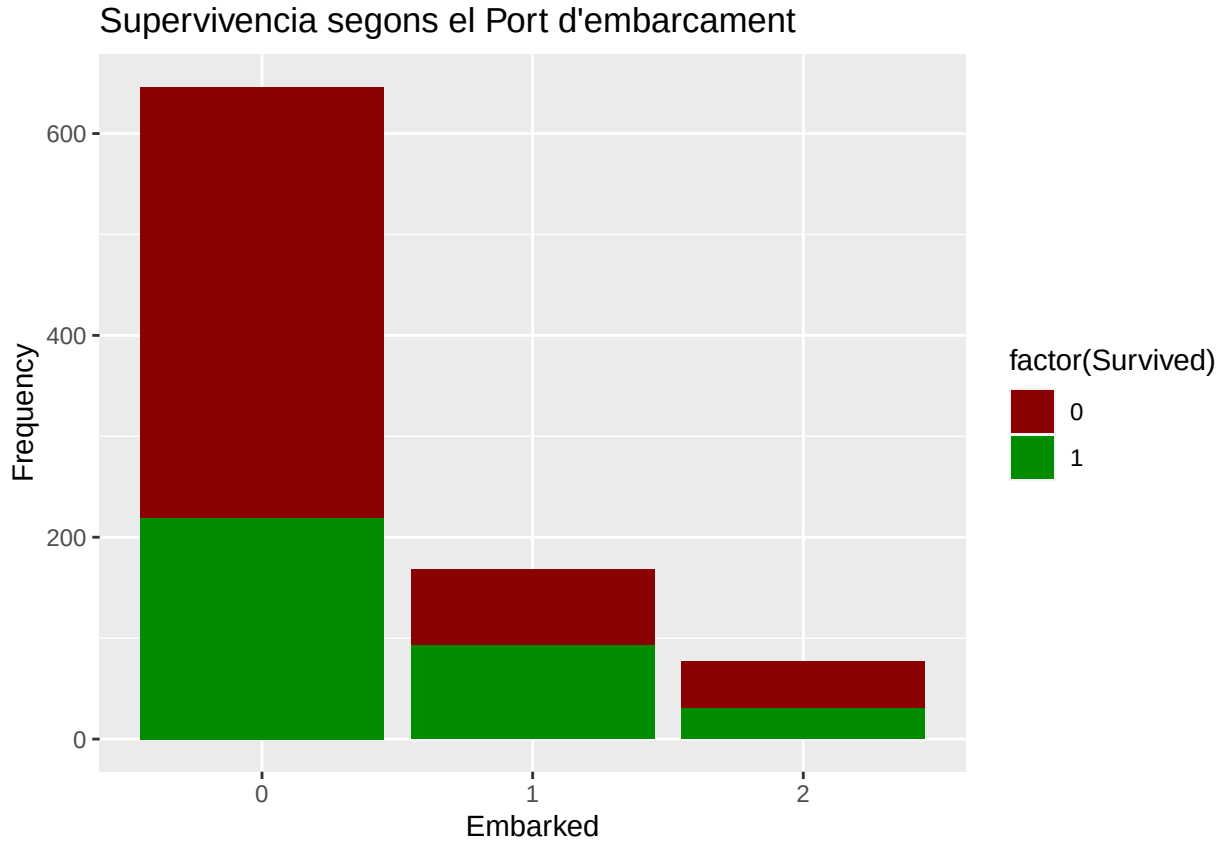
Amb aquests gràfics veiem una forta relació entre la tarifa i la supervivencia dels passatgers. Aquells que van adquirir una tarifa molt més econòmica tenien possibilitats de supervivencia molt menors que les persones que van adquirir tarifes més elevades, que com veiem són molts menys bitllets però amb un índex de supervivencia més elevat.

```r
# Supervivencia per classe del passatge

ggplot(titanic, aes(x=Pclass, fill=factor(Survived))) +
  geom_bar()+
  ylab("Frequency")+
  ggtitle("Supervivencia segons la Classe del passatger")+
  scale_fill_manual(values=c("red4", "green4"))
```



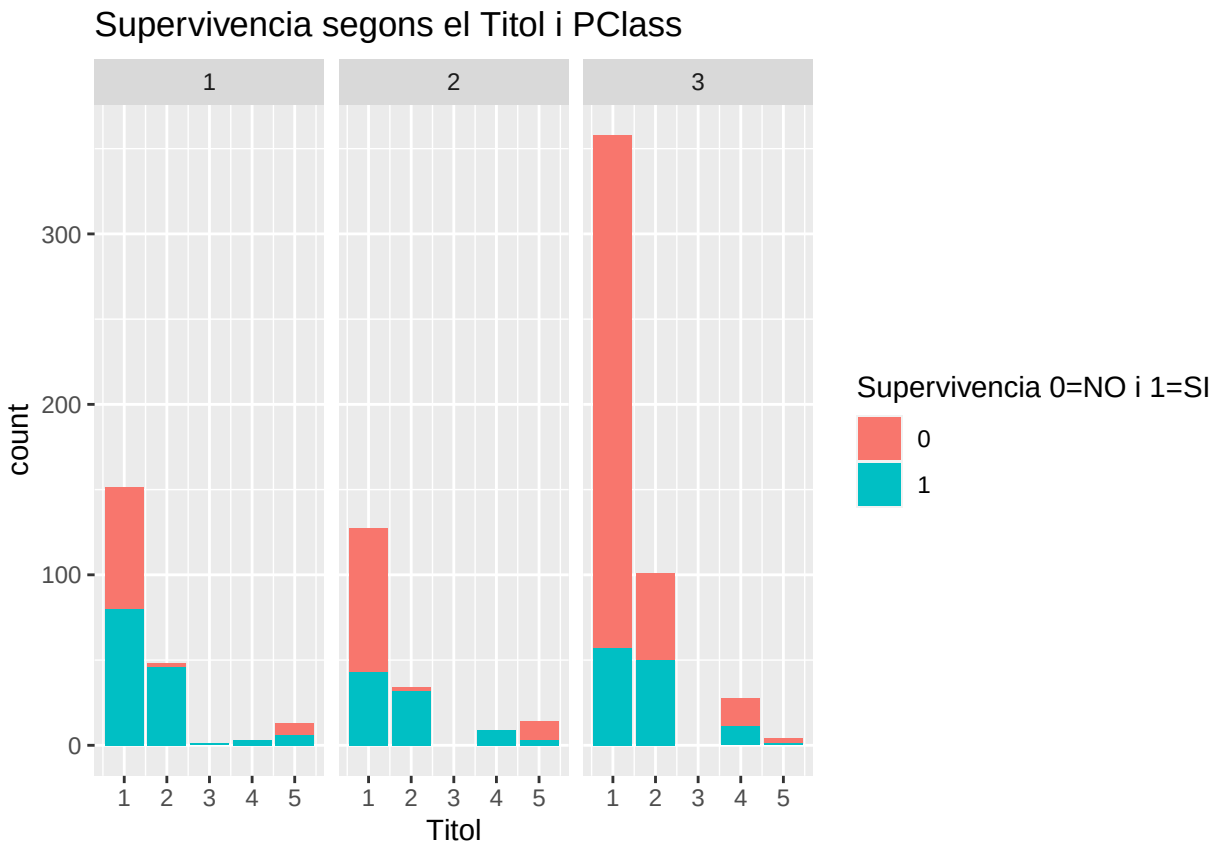
Observem que el nombre de passatgers més elevat eren passatgers de tercera classe, seguit dels de primera classe i a poca distancia dels de segona classe. Tenint en comtpe la quantitat de persones a cada classe, observem que els passatgers a Primera classe tenen un rati de més del 50% de supervivència. Els passatgers de Segona Classe tenen un rati del 50% de supervivència i els passatgers que viatjaven en 3ª classe tenen un rati molt més inferior de supervivència.

```r
# Supervivencia per Port d'embarcament

ggplot(titanic, aes(x=Embarked, fill=factor(Survived))) +
  geom_bar()+
  ylab("Frequency")+
  ggtitle("Supervivencia segons el Port d'embarcament")+
  scale_fill_manual(values=c("red4", "green4"))
```



Segons el resultat de la gràfica podem veure que una molt gran quantitat de passatgers van embarcar al port de Southampton, seguits dels passatgers que van embarcar al port de Cherburgo, i en menor quantitat els que van embarcar al port de Queenstown. Si veiem el grau de supervivencia tenint en compte el nombre de passatgers que van embarcar a cada port, podriem dir que els passatgers que més van sobreviure del total embarcats van ser els del port de Cherburgo.

```
# Supervivencia segons el titol de les persones i la classe on es trobaven allotjats al vaixell

titanic$Pclass<-as.factor(titanic$Pclass)
titanic$Titol<-as.integer(titanic$Titol)

ggplot(titanic, aes(x=Titol, fill=factor(Survived))) +
  geom_bar()+
  facet_grid(.~Pclass)+
  ggtitle("Supervivencia segons el Titol i PClass")+
  scale_fill_discrete(name="Supervivencia 0=NO i 1=SI")
```




```
titanic$Pclass<-as.integer(titanic$Pclass)
titanic$Titol<-as.factor(titanic$Titol)
```

Mitjançant aquesta gràfica voliem veure si les variables titol tenien relació en el grau de supervivencia dels passatgers depenent de la classe on es trobessin. Veient la gràfica fem la següent lectura: - A la categoria pertanyent a Titol 1 (Mr-Don), el que primer veiem és que el nombre de passatgers d'aquest titol a tercera classe, és molt més elevat (més del doble) que en les altres dues classes. El grau de supervivencia es més elevat a primera classe que a segona i a tercera classe.

- De la categoria de Titol 2 (Miss-Mlle-Ms), si ens fixem en la supervivencia només, veiem que a totes 3 classes, aquesta és similar, però si ens fixem en la no supervivencia, veiem que realment apareix només amb un nombre elevat a tercera classe.
- En quant a la categoria 3 i 4 (Mrs-Mme i Master), veiem que a primera classe hi ha un grau de supervivencia màxim, a segona classe també (no hi ha passatgers a la categoria 3), i a tercera classe ens fixem que no hi ha passatgers en categoria 3, però a la 4 tot i que hi ha supervivents, aquests són molt

inferiors (menys del 50%) als no supervivents.

- De la darrera categoria 5 (Don-Dr-Captain..), veiem que la majoria es troben a Primera i Segona classe, i entre aquests dos, el grau de supervivencia és molt més elevat a primera classe que a segona.

6. Resolució del problema.

El principal problema a resoldre és **sapiguer quina probabilitat de sobreviure té un passatger en funció de les seves caracterísques** : clase en la que viatja, edat, sexe, preu del passatge, etc.

Com la variable objecte del estudi (dependent) és dicotòmica en centrem en un model de regressió logística per a resoldre el problema, fent primer un model amb 3 variables regressores (Pclass, Sex i Age) i un altre amb totes les variables regresores posibles.

Ademés com a complement, **ens interessa conèixer si el preu que es paga pel passagte (Fare) és més gran en els passatgers que sobreviuen**. Per a resoldre aquest problema plantegem un contrast de hipòtesis.

Les conclussions obtingudes son:

1. **Els models de regressió logística s'adequan molt bé al cas tractat** donant un accuraccy per sobre de 0.80.
2. La variable **Sex és la que més influència té en qualsevols dels casos.** Les dones tenen més probabilitat de sobreviure que els homes (degut al fet que en els bots salvavides primer van dones i nens.)
3. **Al augmentar el número de variable regressores millora l'adjusts del model**, degut sobretot a que reduim els graus de llibertat i a que incorporem altres variables importants com el Titol i el flag de viatjar sol o no (isAlone).
4. **La variable Titol afegeix molt adjust al model**, ja que millora la importància de Pclass (podriem dir que son dos variables relacionades). En concret el valor 3 (Ms,Mme) va pujar la probabilitat al 0.99.
5. Mitjançant el contrast de hipòtesis de la diferència de mitjanes concluim que **els passatger que sobreviuen paguen un passatge més car.** Acceptem la hipotesis alternativa d'aquest contrast en base al valor de l'estadístic i p-value obtinguts, que en aquest cas és ínfim i per tant menor que el nivell de significació de 0.05.
6. **Utilitzant altres models predictius sembla que estem incorrem en overfitting**, amb el que obtenim models molt adjustats amb el train set però prediccions dolentes amb el test set.
7. **Segons el model de classificació la primera variable discriminadora és Sex**. Després li segueixen Titol (valor 4), Fare i Pclass.

7. Codi

La realització de la pràctica s'ha fet amb R mitjançant el RStudio. El resultat del codi realitzat és el fitxer Practica2.rmd adjuntat a la wiki de Github : https://github.com/jmontero-ob/UOC-PRAC2-Titanic

```
# Creem els dos nous datasets
write.csv(titanic,file="titanic_net.csv")
write.csv(titanic_test,file="titanic_test_net.csv")
```