Pràctica Final

Cristian Subirana

25 de Decembre de 2019

Contents

0.1 Descripció del dataset

El dataset seleccionat per a realitzar aquesta pràctica ha estat el referent al enfonsament del titànic.

El motiu de elecció d'aquest dataset és purament didàctic, juntament amb la possibilitat de participar en el concurs organitzat per Kaggle, on es presenta el repte de crear el millor algoritme de Machine learning capaç de donar els millors resultats alhora de saber qui sobreviu al accident.

L'objectiu d'aquest projecte es aplicar els coneixements assolits durant l'assignatura, juntament amb aconseguir un model capaç de predir per a cada passatger si sobreviu.

0.1.1 Presentació de les dades

En el data set proporcionat per kaggle, podem trobar dos subconjunts de dades, el fitxer train.csv i el fitxer test.csv.

El primer fitxer serà utilitzar per estudiar les dades, tractar-les i formular un model capaç de predir si un passatger sobreviu o no. Un cop finalitzar s'entrenarà el model utilitzant el fitxer train.csv.

El conjunt de variables disponibles en el fitxer traint.csv:

- PassengerId: Identificador numèric únic per identificar el passatger.
- Survived: Flag referent a si el passatger ha sobreviscut o no.
- Pclass: Classe en la que viatjava el passatger.
- Name: Nom del passatger.
- Sex: Sexe del passatger.
- Age: Edat del passatger.
- SibSp: Nombre de germans o acompanyants del passatger.
- Parch: Nombre de pares/fills del passatger.
- Ticket: Número de ticket del passatger.
- Fare: Tarifa del passatger.
- Cabin: Cabina seleccionada pel passatger.
- Embarked: Port on ha embarcat el passatger.

Els camps comentats es veuran afectats al llarg de la pràctica a fi de poder ser analitzats. Inicialment no es descartarà cap variable, ja que tot i que algunes de elles poden no semblar útils en el anàlisi, com per exemple el nom de passatger, poden resultar útils, pel que es mantindran el màxim de variables disponibles.

0.2 Integració i selecció de les dades

Primer de tot, necessitem carregar el dataset. On combinarem les dades de train i test juntes, ja que analitzar-les per separat no te sentit ja que realment són fragments de la mateix font. Les dades de test tenen la mateix estructura que les de train però aquestes no contenen la variable Survived, el qual creem per poder fusionar les dos fonts. Inicialment evaluare'm la columna creada a test amb NA's

```
library(readr)
train_x <- read_csv("C:/Users/PcCom/Desktop/titanic/train.csv")
test_x <- read_csv("C:/Users/PcCom/Desktop/titanic/test.csv")

test_x$Survived<-NA
```

Combinem test i train

```
train<-rbind(train_x,test_x)
```

Observe'm el resultat:

```
str(train)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1309 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  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 12
##   .. ..$ PassengerId: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Survived   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Pclass     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Name       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Sex        : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Age        : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ SibSp      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Parch      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Ticket     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Fare       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ Cabin      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Embarked   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

Un cop ja el tenim carregat, observare'm les dades visualment tal i com són sense aplicar cap procés previ, a fi de tenir una idea general de com són. Per a fer-ho farem les següents consultes:

```
summary(train)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age            SibSp            Parch
```

```
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                     NA's   :263
##     Ticket              Fare            Cabin
##  Length:1309        Min.   :  0.000   Length:1309
##  Class :character   1st Qu.:  7.896   Class :character
##  Mode  :character   Median : 14.454   Mode  :character
##                     Mean   : 33.295
##                     3rd Qu.: 31.275
##                     Max.   :512.329
##                     NA's   :1
##    Embarked
##  Length:1309
##  Class :character
##  Mode  :character
##
##
##
##
```

```
head(train)
```

```
## # A tibble: 6 x 12
##   PassengerId Survived Pclass Name  Sex     Age SibSp Parch Ticket  Fare
##         <int>    <int>  <int> <chr> <chr> <dbl> <int> <int> <chr>  <dbl>
## 1           1        0      3 Brau~ male     22     1     0 A/5 2~  7.25
## 2           2        1      1 Cumi~ fema~    38     1     0 PC 17~ 71.3
## 3           3        1      3 Heik~ fema~    26     0     0 STON/~  7.92
## 4           4        1      1 Futr~ fema~    35     1     0 113803 53.1
## 5           5        0      3 Alle~ male     35     0     0 373450  8.05
## 6           6        0      3 Mora~ male     NA     0     0 330877  8.46
## # ... with 2 more variables: Cabin <chr>, Embarked <chr>
```

Observem que al veure el resum del dataset podem treure algunes conclusions que ens ajudaran a entendre les dades.

La variable candidata a ser exclosa del analisi és el camp Ticket, ja que és un identificador del ticket, dels quals no considero rellevant alhora de decidir qui sobreviu.

Veiem que la mitjana d'edat de les persones que van sobreviure al desastre és del 38%, on majoritariament eren persones que viatjaven amb classe mitja, amb una mitjana d'edat d'uns 30 anys, dels quals majoritariament viatjaven sols.

Cal considerar que a la variable Age,Fare són les única variables que contenen NA's pel que en el següents apartats s'hauran de tractar.

Al observar el resultat de fer un head() de les dades veiem que hi ha un conjunt de camps que requeriran d'un treball extra tant de anàlisis com de transformació.

Variables com Cabin, veiem que estan definides com una lletra i un nombre referent a la cabina, on la lletra fa referència a quina altura estava la cabina.

Observe'm la següent imatge per a tindre'n una idea de com estaven distribuïdes aquestes cabines per tal de saber la importància de la lletra que conté.

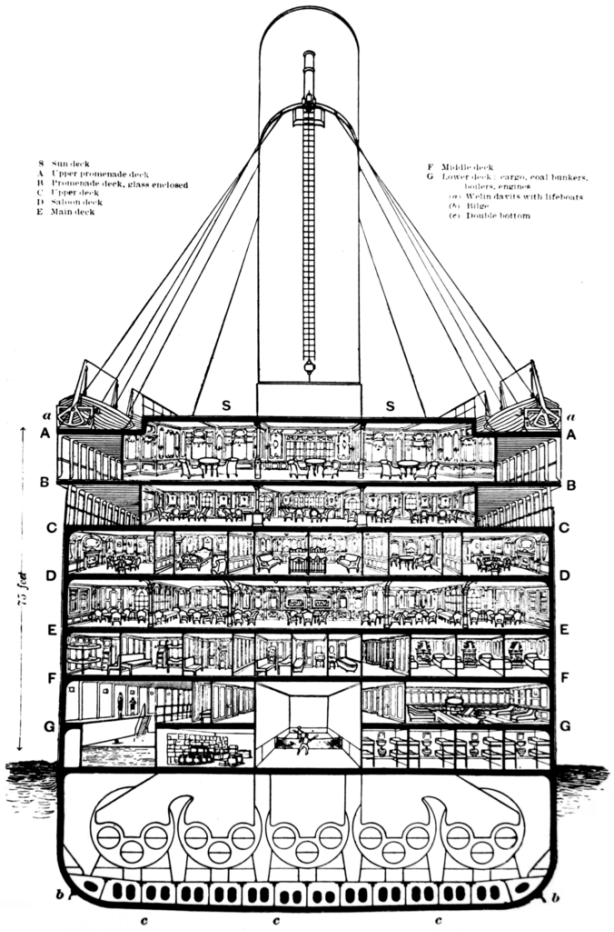


Figure 1: Titanic

Observe'm en la imatge, que a major lletra, més aprop del bots salvavides es troben i millor accés a la superfície, per tant, pot ser un factor a tenir en compte alhora de saber si sobreviu o no.

El camp referent a Embarked, fa referència als ports des de els quals els passatgers van embarcar, sent C=Cherbourg, Q=Queenstown i S=Southampton.

En la següent imatge podem veure les embarcacions presents per pujar al titanic

Figure 2: Embarcacions

0.3 Neteja de les dades

En aquest apartat tractarem les variables del dataset, per tal de gestionar els nulls, NA's i valors buits. Acte seguit es preparan les variables per a poder ser analitzades correctament.

En el últim head() utilitzat podem apreciar que la variable Cabin, tot i que al fer summary() de dataset no ens ha sortit que contingui NA's, apreciem clarament que en conté.

El primer pas que farem és eliminar la variable Ticket del anàlisi tal i com s'ha explicat anteriorment

```
train_c1<-train[,-which(names(train)=="Ticket")]
```

Comprove'm que ja no hi és:

```
summary(train_c1)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age            SibSp            Parch
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                     NA's   :263
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:1309        Length:1309
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 33.295
##  3rd Qu.: 31.275
##  Max.   :512.329
##  NA's   :1
```

0.3.1 Variables PassengerId i Survived

Aquestes dos variables no seran transformades degut a que PassengerId ens permetrà en tot moment identificar el passatger i Survived és la variable sobre la que volem realitzar l'anàlisi. Hem de tenir en compte que per poder pujar el resultat del projecte a Kaggle es requereixen de només dues columnes, PassengerId i Survived.

0.3.2 Variable Name

Aquesta variable contenia el nom del passatger. Observe'm els seus valors.

```
head(train_c1$Name)
```

```
## [1] "Braund, Mr. Owen Harris"
## [2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## [3] "Heikkinen, Miss. Laina"
## [4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
## [5] "Allen, Mr. William Henry"
## [6] "Moran, Mr. James"
```

Veiem que aquest camp, pot semblar poc important, ja que el passatger es digui Pep o Marc difícilment influenciarà si el supervivent sobreviu.

Aquest camp també conté el títol amb el que ens dirigiríem al passatger, el qual podem entendre que pot ser el estatus o nivell social del passatger, el qual sí que pot influenciar en sí sobreviu.

Per tant, hem de netejar aquest camp, deixant només el que vindria a ser el titol.

Separe'm els noms per caracters com comes o punts i seleccionem únicament el títol

```
library(stringr)
train_c1$Title<-sapply(train_c1$Name,FUN = function(x) str_trim(unlist(strsplit(x,split='[,|.]'))[2]))
```

Observe'm el resultat de netejar la variable Name:

```
table(train_c1$Title)
```

```
##
##         Capt          Col          Don         Dona           Dr
##            1            4            1            1            8
##     Jonkheer         Lady        Major       Master         Miss
##            1            1            2           61          260
##         Mlle          Mme           Mr          Mrs           Ms
##            2            1          757          197            2
##          Rev          Sir the Countess
##            8            1            1
```

Veiem que podríem dividir els titols segons si el passatger es una dona, noia, home o noi per exemple.

Primerament hem de tenir en compte el següent:

Mr->Home adult, independentment del seu estat civil Mrs->Dona casada Miss->Dona soltera jove Ms->Dona adulta, independement del seu estat civil Sir->Home de classe alta o distinguit per la seva profesió o conducta, respecte o cortesia Mme->Dona de classe alta o distinguida per la seva profesió o conducta, respecte

o cortesia, equivaldria a Madam Sir the Countess->Home amb carrec molt important, compte Lady->Forma aducada de dirigirse a una dona Capt->Capità, mariner com a professió Major->carrec molt important dins d'una area Master-> Per referirse a nens Rev->Sacerdot Dr-> Doctor en medicina Mlle->Equivalent de mademoiselle o de Miss Jonkheer->titol nobiliari especific de la familia comentada Col<-coronel The countess<- compte

Observe'm que hi han equivalències en les definicions el qual poden ser degudes a que la majoria de tripulants provenien de França, Regne Unit i els Estats Units principalment, i per referir-nos al mateix, degut a l'idioma s'escriuen diferent.

Per a verificar els titols, s'han comprovat dels passatgers, quina funció tenien dins del titanic a https://www.encyclopedia-titanica.org/

Gràcies a aquesta verificació s'ha comprovat que el dataset no inclou els treballadors del vaixell,en que el passatger amb títol capità, no era el capità del vaixell, sinó que era capità d'un altre vaixell, i estava de viatje turístic al titanic.

Un cop comentat el següent es faran les següents agrupacions:

Nens<-Master Dona soltera<- Miss,Mlle Dona casada<-Mrs Dona sense estat civil<-Ms,Lady Home sense estat civil<-Mr,Sir Home de classe alta<-Sir the Countess,Capt,Major,Jonkheer,Don,Col Dona de classe alta<-Mme Sacerdot<-Rev Doctors<-Dr

Prepare'm variables

```
nens<-c("Master")
dona_soltera<-c("Miss","Mlle")
dona_casada<-c("Mrs")
dona_sense_estat_civil<-c("Ms","Lady")
home_sense_estat_civil<-c("Mr","Sir")
home_clase_alta<-c("the Countess","Capt","Major","Jonkheer","Don","Dona","Col")
dona_clase_alta<-c("Mme")
sacerdot<-c("Rev")
doctor<-c("Dr")
```

Creem una nova variable al dataset

```
train_c1$Title_refactor<-vector(mode="character",length = nrow(train_c1))
```

Afegim les dades al nou camp segons el tipus de títol especificat anteriorment

```
train_c1$Title_refactor[train_c1$Title %in% nens]<-"Nens"
train_c1$Title_refactor[train_c1$Title %in% dona_soltera]<-"Dona soltera"
train_c1$Title_refactor[train_c1$Title %in% dona_casada]<-"Dona casada"
train_c1$Title_refactor[train_c1$Title %in% dona_sense_estat_civil]<-"Dona sense estat civil"
train_c1$Title_refactor[train_c1$Title %in% home_sense_estat_civil]<-"Home sense estat civil"
train_c1$Title_refactor[train_c1$Title %in% home_clase_alta]<-"Home clase alta"
train_c1$Title_refactor[train_c1$Title %in% dona_clase_alta]<-"Dona clase alta"
train_c1$Title_refactor[train_c1$Title %in% sacerdot]<-"Sacerdot"
train_c1$Title_refactor[train_c1$Title %in% doctor]<-"Doctor"
```

Test de la nova variable

```
head(train_c1)
```

```
## # A tibble: 6 x 13
##   PassengerId Survived Pclass Name  Sex     Age SibSp Parch  Fare Cabin
##         <int>    <int>  <int> <chr> <chr> <dbl> <int> <int> <dbl> <chr>
## 1           1        0      3 Brau~ male     22     1     0  7.25 <NA>
## 2           2        1      1 Cumi~ fema~    38     1     0 71.3  C85
## 3           3        1      3 Heik~ fema~    26     0     0  7.92 <NA>
## 4           4        1      1 Futr~ fema~    35     1     0 53.1  C123
## 5           5        0      3 Alle~ male     35     0     0  8.05 <NA>
## 6           6        0      3 Mora~ male     NA     0     0  8.46 <NA>
## # ... with 3 more variables: Embarked <chr>, Title <chr>,
## #   Title_refactor <chr>
```

Test de la nova variable

```
unique(train_c1$Title_refactor)
```

```
## [1] "Home sense estat civil" "Dona casada"
## [3] "Dona soltera"           "Nens"
## [5] "Home clase alta"        "Sacerdot"
## [7] "Doctor"                 "Dona clase alta"
## [9] "Dona sense estat civil"
```

0.3.3 Variable Age

A continuació la variable a estudiar és "Age", el qual havíem vist que conté NA's.

Per a trobar una solució, tenim varis camins pels quals optar:

1. Substituir els NA's per les mitjanes de edat.
2. Substituir els NA's per la mitjana de edat per cada títol.
3. Aplicar kNN per a substituir els NA's

Crec que la opció 3 seria la més idonea, ja que en la opció 1, pot apareixer un important esbiaix. La opció 2 també seria viable, però la opció de utilitzar knn és la més recomanada especialment per a variables numériques.

Primerament per a que el knn sigui òptim, normalitzem la variable Age

```
train_c1$Age<-scale(train_c1$Age)
```

Apliquem knn per a substituir els NA's

```
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## Loading required package: data.table
```

```
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.

## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
x<-kNN(train_c1,variable = c("Age"),k=5)
train_c1<-x
```

Revise'm resultat

```
summary(train_c1)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age              SibSp            Parch
##  Length:1309        Min.   :-2.06123   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:-0.61616   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :-0.19991   Median :0.0000   Median :0.000
##                     Mean   :-0.03394   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.: 0.45915   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   : 3.47694   Max.   :8.0000   Max.   :9.000
##
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:1309        Length:1309
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 33.295
##  3rd Qu.: 31.275
##  Max.   :512.329
##  NA's   :1
##     Title           Title_refactor      Age_imp
##  Length:1309        Length:1309        Mode :logical
##  Class :character   Class :character   FALSE:1046
##  Mode  :character   Mode  :character   TRUE :263
##
##
##
##
```

Finalment veiem que ja no hi han NA's.

0.3.4 Variable Embarked

En la variable Embarked observe'm que hi han NA's pel que hem de decidir com tractar-los.

```
unique(train_c1$Embarked)
```

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

Al ser variables categòriques, ens podem basar en la categoria més abundant, el qual substituirà els NA's.

Busquem quina categoria és la més habitual.

```
xtabs(~Embarked,data=train_c1)
```

```
## Embarked
##   C   Q   S
## 270 123 914
```

Veiem que el embarcament majoritari és S.

```
train_c1$Embarked[is.na(train_c1$Embarked)]<-'S'
```

Revise'm que ja no apareixin més NA's al camp.

```
unique(train_c1$Embarked)
```

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

0.3.5 Variable Cabin

La variable Cabin en el dataset de training conté 1014 NA's

```
length(which(is.na(train_c1$Cabin)))
```

```
## [1] 1014
```

Hem de tenir en compte que en aquesta variable, no tothom tenia cabina associada, pel que el volum de cabines ocupades no coincidirà amb el nombre de passatgers, a més que hi havia passatgers que compartien cabina.

El que ens interessaria d'aquesta variable és la primera lletra que conté la variable, ja que aquesta referencía al pis/bloc dins del vaixell, on anteriorment hem mostrat una imatge amb la distribució de les cabines. L'ordre de les cabines anava des de les lletres A fins a Z

```
unique(substr(train_c1$Cabin,1,1))
```

```
## [1] NA  "C" "E" "G" "D" "A" "B" "F" "T"
```

Observe'm que hi ha un valor que desconeixem, el valor "T". Si ens fixem en la dsitribució de les cabines per la lletra de la imatge anterior, veiem que no hi apareix cap T, tot i que realment existia. Podem trobar-la a https://www.encyclopedia-titanica.org/titanic-deckplans/boat-deck.html

Si observe'm el vaixell des de una vista superior, veiem que hi ha una cabina adalt de tot, el qual és única que fa referència aquest "T", pel que podríem considerar que la situació de la cabina T és més propera als bots salvavides que les cabines "A".

Un cop finalitzat aquest estudi, modificarem el camp cabina, deixant només la lletra de la cabina a la que fa referència. En el cas dels NA's se'ls assignarà la lletra Z, com els menys accesibles als bots.

```
train_c1$Cabin[is.na(train_c1$Cabin)]<-'Z'
train_c1$Cabin<-substr(train_c1$Cabin,1,1)
```

Comprove'm el resultat

```
unique(substr(train_c1$Cabin,1,1))
```

```
## [1] "Z" "C" "E" "G" "D" "A" "B" "F" "T"
```

0.3.6 Variable Pclass

Creem una nova variable al dataset

```
train_c1$classe<-vector(mode="character",length = nrow(train_c1))
```

Afegim les dades al nou camp segons el tipus de titol especificat anteriorment

```
train_c1$classe[train_c1$Pclass==1]<-"Alta"
train_c1$classe[train_c1$Pclass==2]<-"Mitja"
train_c1$classe[train_c1$Pclass==3]<-"Baixa"
```

```
unique(train_c1$classe)
```

```
## [1] "Baixa" "Alta"  "Mitja"
```

0.3.7 Variable SibSp i Parch

Aquestes variables les deixem tal i com estan.

0.3.8 Variable Fare

Degut a que aquesta variable conté NA's hem de realitzar un procés semblant al realitzat per la variable Age.

```
train_c1$Fare<-scale(train_c1$Fare)
```

Apliquem knn per a substituir els NA's

```
library(VIM)
x<-kNN(train_c1,variable = c("Fare"),k=5)
train_c1<-x
```

Revise'm resultat

```
summary(train_c1)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age               SibSp            Parch
##  Length:1309        Min.   :-2.06123   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:-0.61616   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :-0.19991   Median :0.0000   Median :0.000
##                     Mean   :-0.03394   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.: 0.45915   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   : 3.47694   Max.   :8.0000   Max.   :9.000
##
##       Fare              Cabin             Embarked
##  Min.   :-0.643283   Length:1309        Length:1309
##  1st Qu.:-0.490733   Class :character   Class :character
##  Median :-0.364022   Mode  :character   Mode  :character
##  Mean   :-0.000376
##  3rd Qu.:-0.039037
##  Max.   : 9.255140
##
##     Title           Title_refactor      Age_imp           classe
##  Length:1309        Length:1309        Mode :logical   Length:1309
##  Class :character   Class :character   FALSE:1046      Class :character
##  Mode  :character   Mode  :character   TRUE :263       Mode  :character
##
##
##
##
##   Fare_imp
##  Mode :logical
##  FALSE:1308
##  TRUE :1
##
##
##
##
```

Cal tenir en compte que hi han valors de ticket=0, el qual pot semblar un outlier, però el motiu d'aquest valor, és degut a que en el titanic, per un mateix ticket podien entrar N passatgers, pel que el valor del ticket va associat al passatger que el va pagar, però els seus acompanyants apareix amb valor 0. Un acompanyant pot no ser familiar ni parella, pot ser per exemple amics o "nanny's".

Netejem les variables que no necessitem finalment

```
train_c1<-train_c1[,-which(names(train_c1)=="Title")]
```

```
train_c1<-train_c1[,-which(names(train_c1)=="Fare_imp")]
```

```
train_c1<-train_c1[,-which(names(train_c1)=="Age_imp")]
```

```
head(train_c1)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex        Age
## 1                            Braund, Mr. Owen Harris   male -0.5467832
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  0.5632089
## 3                             Heikkinen, Miss. Laina female -0.2692852
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  0.3550854
## 5                           Allen, Mr. William Henry   male  0.3550854
## 6                                   Moran, Mr. James   male -0.6161577
##   SibSp Parch       Fare Cabin Embarked         Title_refactor classe
## 1     1     0 -0.5032100     Z        S Home sense estat civil  Baixa
## 2     1     0  0.7339412     C        C            Dona casada   Alta
## 3     0     0 -0.4901687     Z        S           Dona soltera  Baixa
## 4     1     0  0.3826320     C        S            Dona casada   Alta
## 5     0     0 -0.4877536     Z        S Home sense estat civil  Baixa
## 6     0     0 -0.4798651     Z        Q Home sense estat civil  Baixa
```

0.3.9 Dataset Final

Prepare'm el nou dataset:

```
train_net<-train_c1[,-which(names(train_c1) %in% c("Pclass","Name"))]
```

```
summary(train_net)
```

```
##   PassengerId      Survived          Sex                 Age
##  Min.   :   1   Min.   :0.0000   Length:1309        Min.   :-2.06123
##  1st Qu.: 328   1st Qu.:0.0000   Class :character   1st Qu.:-0.61616
##  Median : 655   Median :0.0000   Mode  :character   Median :-0.19991
##  Mean   : 655   Mean   :0.3838                      Mean   :-0.03394
##  3rd Qu.: 982   3rd Qu.:1.0000                      3rd Qu.: 0.45915
##  Max.   :1309   Max.   :1.0000                      Max.   : 3.47694
##                 NA's   :418
##      SibSp            Parch            Fare              Cabin
##  Min.   :0.0000   Min.   :0.000   Min.   :-0.643283   Length:1309
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:-0.490733   Class :character
```

```
##  Median :0.0000   Median :0.000   Median :-0.364022   Mode  :character
##  Mean   :0.4989   Mean   :0.385   Mean   :-0.000376
##  3rd Qu.:1.0000   3rd Qu.:0.000   3rd Qu.:-0.039037
##  Max.   :8.0000   Max.   :9.000   Max.   : 9.255140
##
##    Embarked         Title_refactor        classe
##  Length:1309        Length:1309        Length:1309
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
#str(train_net)
train_net$Sex<-as.factor(train_net$Sex)
train_net$Cabin<-as.factor(train_net$Cabin)
train_net$Embarked<-as.factor(train_net$Embarked)
train_net$Title_refactor<-as.factor(train_net$Title_refactor)

train_net$classe<-as.factor(train_net$classe)


train_net$Survived<-as.factor(train_net$Survived)
```

0.4 Anàlisi de les dades

0.4.1 Test de normalitat

Degut a que el dataset ha estat preparat o orientat a fer ús de regressió, on la majoria de variables no són numériques, utilitzarem el dataset antic, on no hi han NA's, és a dir, que s'ha fet un tractament de les dades, per a estudiar la normalitat de les variables numériques.

En aquest cas les úniques variables que tindria sentit fer analisis de normalitat i homocedasticitat serien les que tenen sentit numèric com Age i Fare.

```
summary(train_c1)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age               SibSp            Parch
##  Length:1309        Min.   :-2.06123   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:-0.61616   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :-0.19991   Median :0.0000   Median :0.000
##                     Mean   :-0.03394   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.: 0.45915   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   : 3.47694   Max.   :8.0000   Max.   :9.000
```

```
##
##       Fare              Cabin             Embarked
##  Min.   :-0.643283   Length:1309        Length:1309
##  1st Qu.:-0.490733   Class :character   Class :character
##  Median :-0.364022   Mode  :character   Mode  :character
##  Mean   :-0.000376
##  3rd Qu.:-0.039037
##  Max.   : 9.255140
##
##  Title_refactor        classe
##  Length:1309        Length:1309
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

Per a fer el test de normalitat faré servir el test de Shapiro.

```
shapiro.test(train_c1$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  train_c1$Age
## W = 0.97263, p-value = 4.805e-15
```

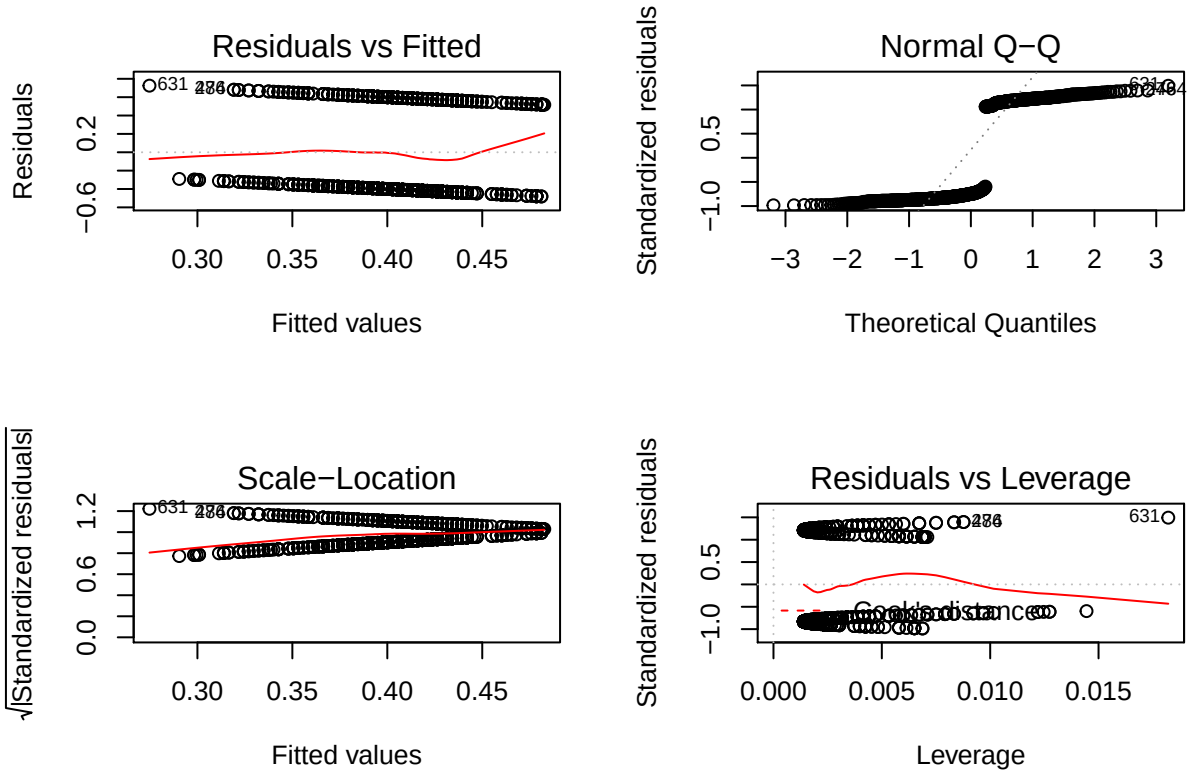
```
shapiro.test(train_c1$Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  train_c1$Fare
## W = 0.52765, p-value < 2.2e-16
```

El resultat del test indica que cap de les variables numériques està normalitzada, ja que els seus p-valors són inferiors a 0.05, pel que podem rebutjar la hipòtesi nul · la, sent les variables estudiades amb una distribució no-normal.

A continuació n'estudiem l'homocedasticitat per la variable Age

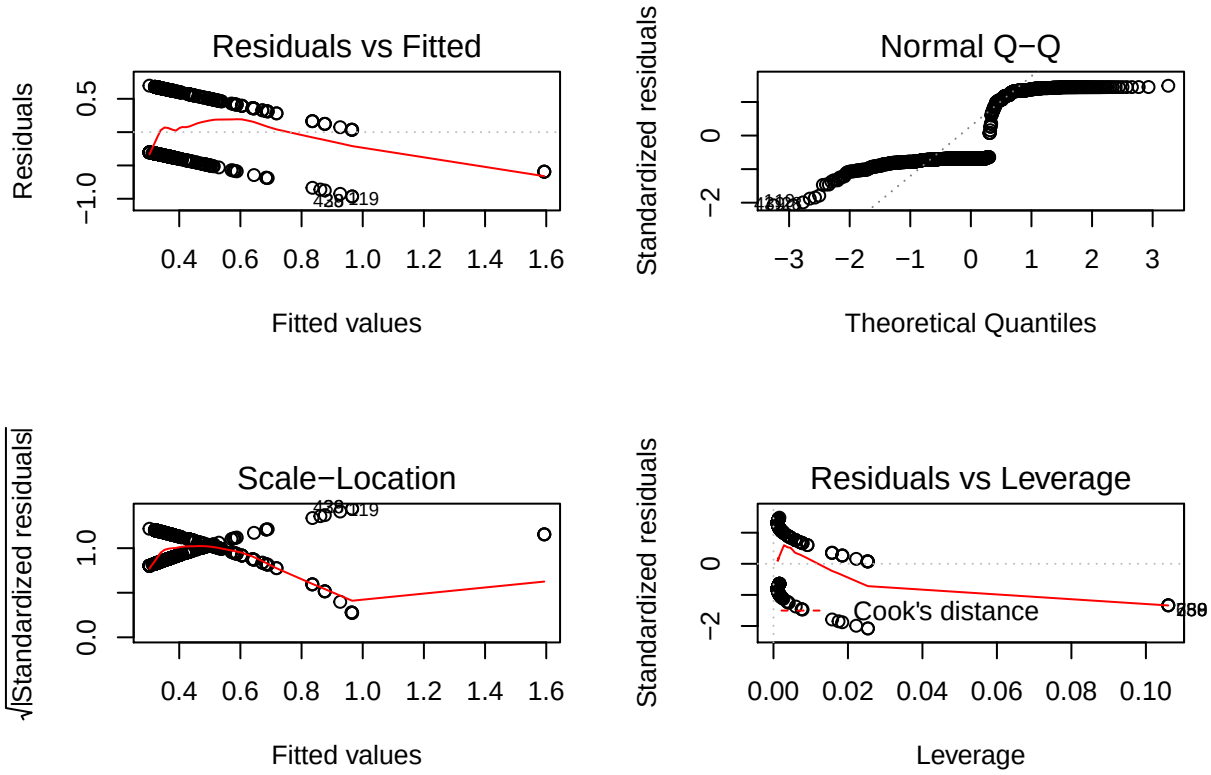
```
lmMod<-lm(Survived~Age,data=train_x)
par(mfrow=c(2,2))
plot(lmMod)
```



Observe'm en el plot de dalt a l'esquerre que la curva vermella no es manté estable, on veiem que els residus semblen augmentar a la vegada que y o fa. Per tant aquesta variable presenta heterocedasticitat

A continuació n'estudiem l'homocedasticitat per la variable Fare

```
lmMod<-lm(Survived~Fare,data=train_x)
par(mfrow=c(2,2))
plot(lmMod)
```



Observe'm en el plot de dalt a l'esquerre que la curva vermella no es manté estable, on veiem que els residus semblen augmentar a la vegada que y o fa. Per tant aquesta variable presenta heterocedasticitat

Degut a que cap de les variables estudiades anteriorment presenta normalitat utilitzaré el test de fligner, el qual és el més comú per a casos on no es compleixen la condició de normalitat.

```
fligner.test(Survived~Age,data=train_c1)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Age
## Fligner-Killeen:med chi-squared = 78.001, df = 87, p-value =
## 0.7442
```

```
fligner.test(Survived~Fare,data=train_c1)
```

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

Observe'm que per les dues variables el p-value és major de 0.05, pel que podem acceptar la hipòtesi nul · la d'homoscedasticitat concloen que aquestes dues variables presenten variancies estadísticament iguals per als seus respectius grups.

0.4.2 Distribució de les variables segons si sobreviu

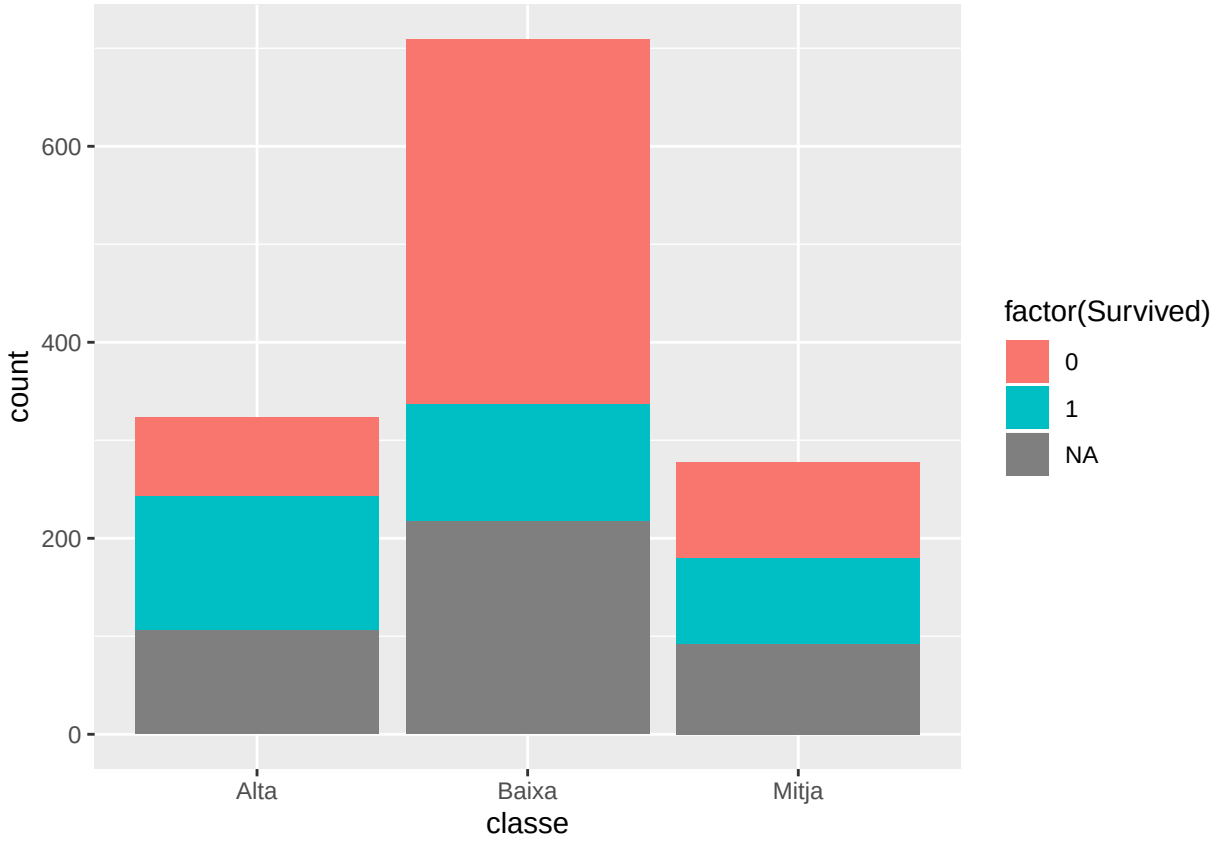
A continuació volem veure per cada variable, els volums de passatgers segons si sobreviu o no, a fi de tenir una visió general de l'importància de cada variable.

0.4.2.1 Factor sobreviu segons classe

```
library(ggplot2)

train_net%>%ggplot(aes(x=classe,fill=factor(Survived)))+geom_bar(stat="count",positin="fill")
```

```
## Warning: Ignoring unknown parameters: positin
```

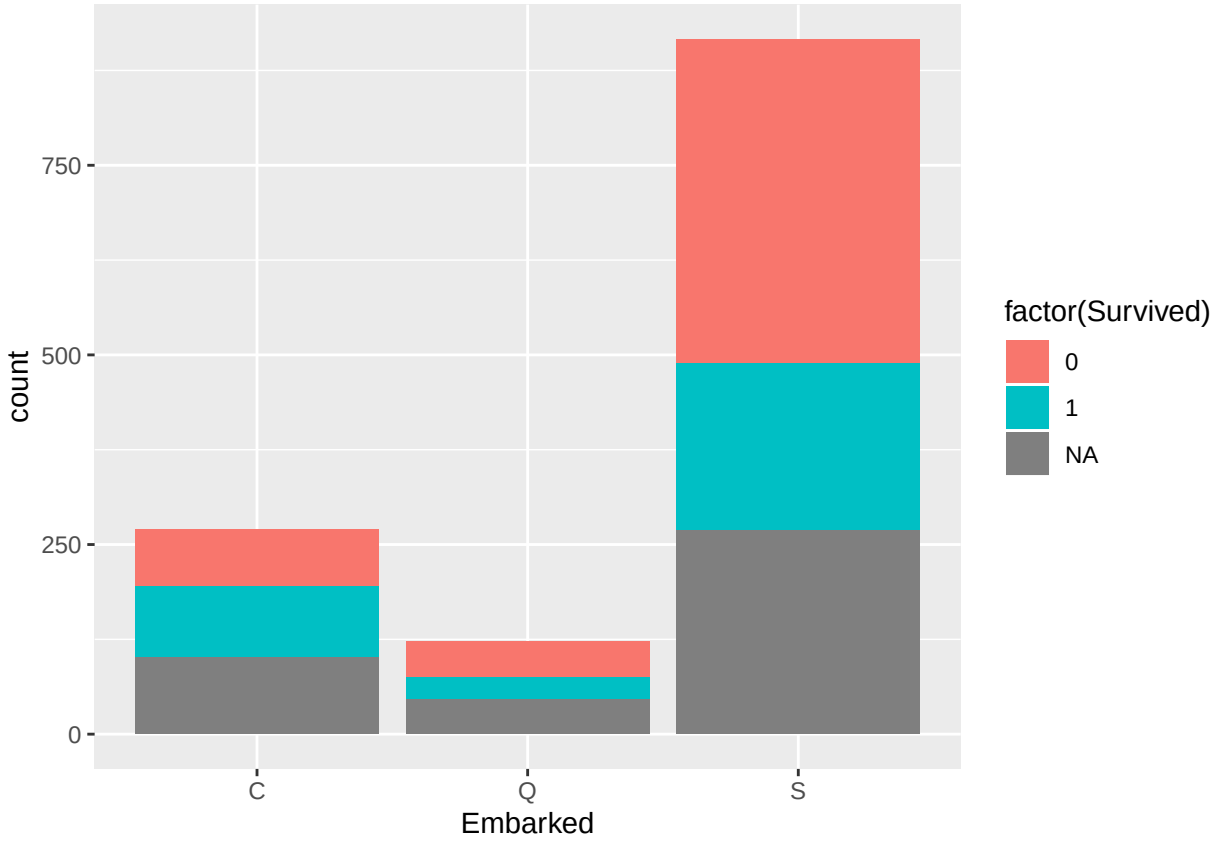


0.4.2.2 Factor sobreviu segons Port d'embarcació

```
library(ggplot2)

train_net%>%ggplot(aes(x=Embarked,fill=factor(Survived)))+geom_bar(stat="count",positin="fill")
```

```
## Warning: Ignoring unknown parameters: positin
```

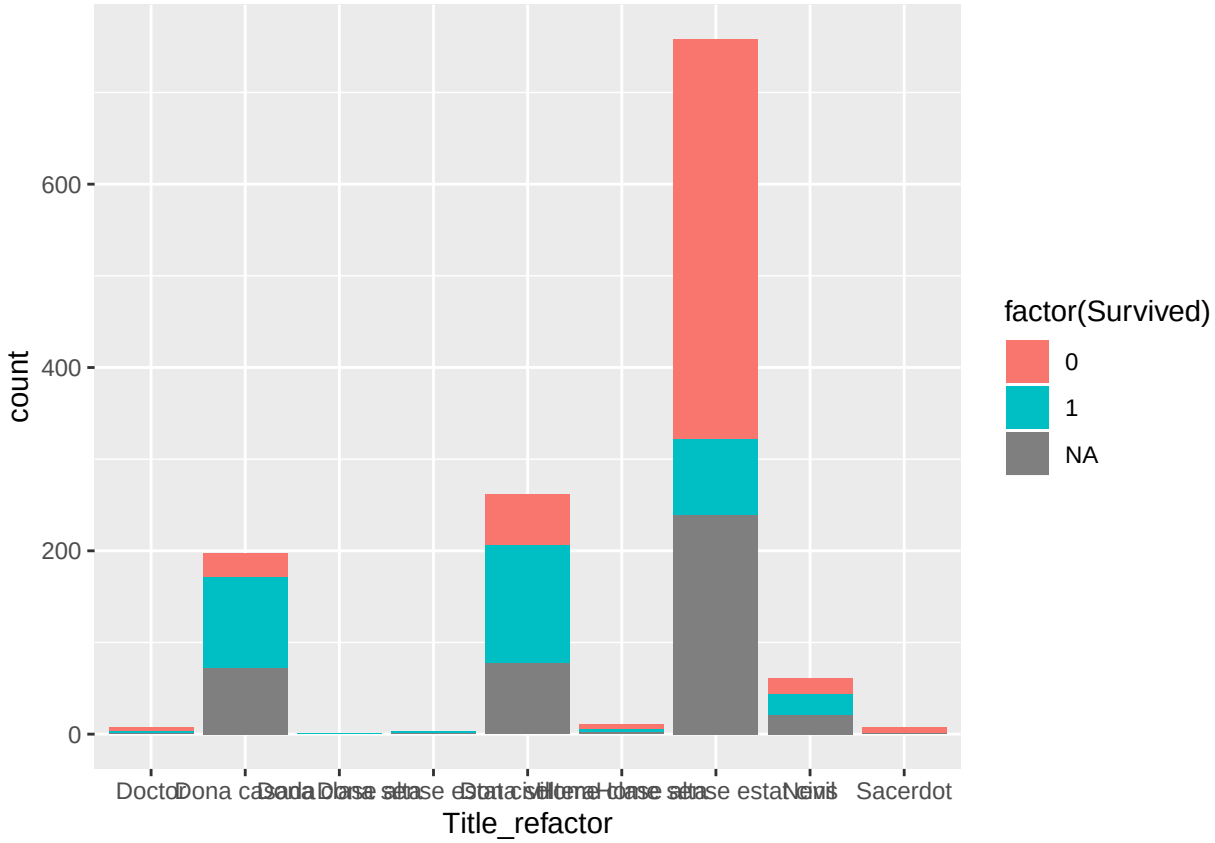


0.4.2.3 Factor sobreviu segons Títol

```
library(ggplot2)

train_net%>%ggplot(aes(x=Title_refactor,fill=factor(Survived)))+geom_bar(stat="count",positin="fill")
```

```
## Warning: Ignoring unknown parameters: positin
```



Per a veure correctament aquest gràfic es requereix de mostrar en una nova finestra a pantalla completa.

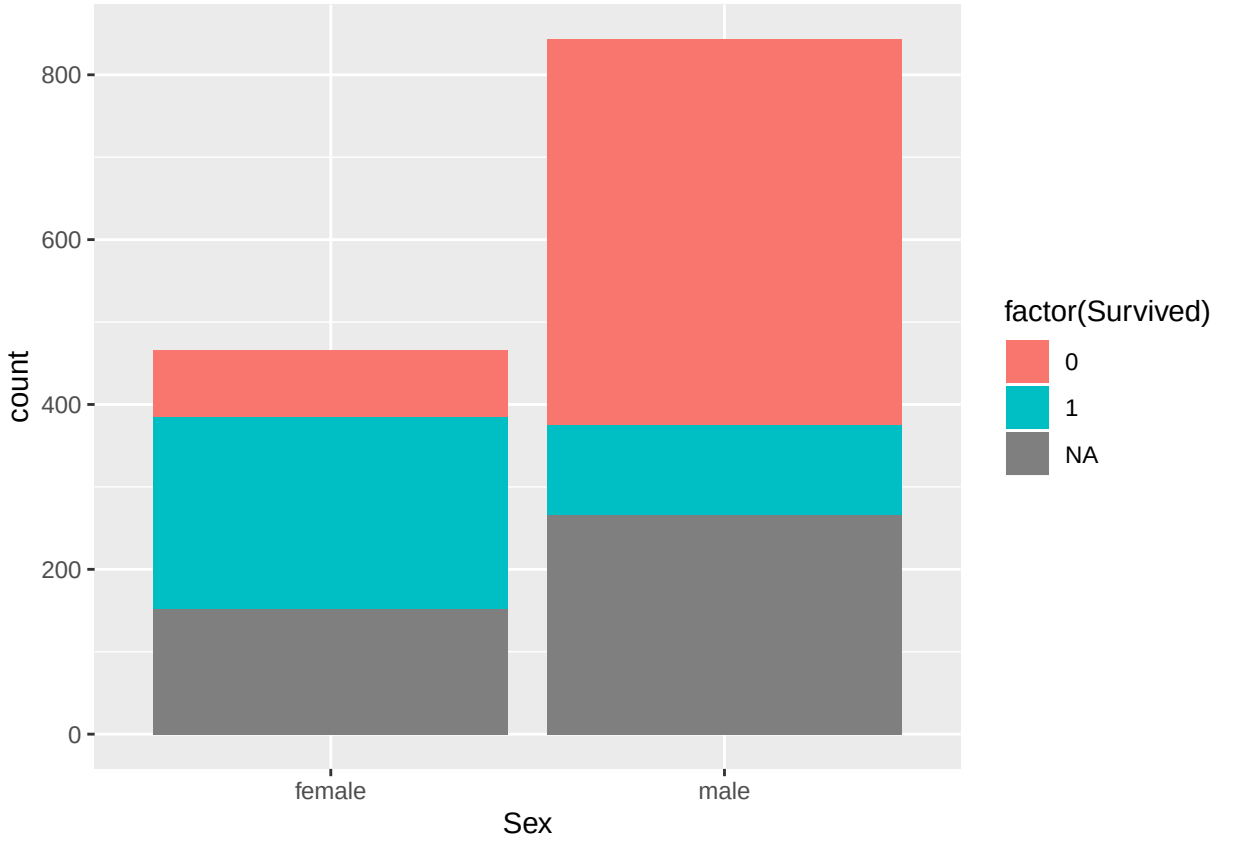
Podem apreciar que ser "Home sense estat civil" pot ser un clar factor de no sobreviure.

0.4.2.4 Factor sobreviu segons Sexe

```
library(ggplot2)

train_net%>%ggplot(aes(x=Sex,fill=factor(Survived)))+geom_bar(stat="count",positin="fill")
```

```
## Warning: Ignoring unknown parameters: positin
```



A continuació apliquem el test de chi quadrat, per veure la significancia entre les variables i la variable Survived

```
chisq.test(train_net$Survived,train_net$Sex)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  train_net$Survived and train_net$Sex
## X-squared = 260.72, df = 1, p-value < 2.2e-16
```

```
chisq.test(train_net$Survived,train_net$Cabin)
```

```
## Warning in chisq.test(train_net$Survived, train_net$Cabin): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  train_net$Survived and train_net$Cabin
## X-squared = 99.164, df = 8, p-value < 2.2e-16
```

```
chisq.test(train_net$Survived,train_net$Embarked)
```

```
##
##  Pearson's Chi-squared test
##
## data:  train_net$Survived and train_net$Embarked
## X-squared = 25.964, df = 2, p-value = 2.301e-06
```

```
chisq.test(train_net$Survived,train_net$Title_refactor)
```

```
## Warning in chisq.test(train_net$Survived, train_net$Title_refactor): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  train_net$Survived and train_net$Title_refactor
## X-squared = 292.56, df = 8, p-value < 2.2e-16
```

```
chisq.test(train_net$Survived,train_net$classe)
```

```
##
##  Pearson's Chi-squared test
##
## data:  train_net$Survived and train_net$classe
## X-squared = 102.89, df = 2, p-value < 2.2e-16
```

Degut a que totes les variables tenen un p-value inferior a 0.05 podem assegurar que totes són significatives per al anàlisi

0.4.3 Model de regressió per a poder predir supervivents

El primer model que utilitzaré és el de randomforest, on primarement dividiré el dataset en dos parts, els que tenen els supervivents informats i els que no.

```
train_clean<-train_net[!is.na(train_net$Survived),]
test_clean<-train_net[is.na(train_net$Survived),]
```

Revise'm que el subdataset estigui bé

```
str(train_clean)
```

```
## 'data.frame':    891 obs. of  11 variables:
##  $ PassengerId   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived      : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Sex           : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age           : num  -0.547 0.563 -0.269 0.355 0.355 ...
##  $ SibSp         : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch         : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare          : num  -0.503 0.734 -0.49 0.383 -0.488 ...
##  $ Cabin         : Factor w/ 9 levels "A","B","C","D",..: 9 3 9 3 9 9 5 9 9 9 ...
##  $ Embarked      : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  $ Title_refactor: Factor w/ 9 levels "Doctor","Dona casada",..: 7 2 5 2 7 7 7 8 2 2 ...
##  $ classe        : Factor w/ 3 levels "Alta","Baixa",..: 2 1 2 1 2 2 1 2 2 3 ...
```

```r
str(test_clean)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId   : int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Survived      : Factor w/ 2 levels "0","1": NA NA NA NA NA NA NA NA NA NA ...
##  $ Sex           : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age           : num  0.32 1.188 2.228 -0.2 -0.547 ...
##  $ SibSp         : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch         : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare          : num  -0.492 -0.508 -0.456 -0.476 -0.406 ...
##  $ Cabin         : Factor w/ 9 levels "A","B","C","D",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ Embarked      : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
##  $ Title_refactor: Factor w/ 9 levels "Doctor","Dona casada",..: 7 2 7 7 2 7 5 7 2 7 ...
##  $ classe        : Factor w/ 3 levels "Alta","Baixa",..: 2 2 3 2 2 2 2 3 2 2 ...
```

```r
str(train_clean$Title_refactor)
```

```
##  Factor w/ 9 levels "Doctor","Dona casada",..: 7 2 5 2 7 7 7 8 2 2 ...
```

```r
unique(train_clean$Title_refactor)
```

```
## [1] Home sense estat civil Dona casada            Dona soltera
## [4] Nens                   Home clase alta        Sacerdot
## [7] Doctor                 Dona clase alta        Dona sense estat civil
## 9 Levels: Doctor Dona casada Dona clase alta ... Sacerdot
```

```r
str(test_clean$Title_refactor)
```

```
##  Factor w/ 9 levels "Doctor","Dona casada",..: 7 2 7 7 2 7 5 7 2 7 ...
```

```r
unique(test_clean$Title_refactor)
```

```
## [1] Home sense estat civil Dona casada            Dona soltera
## [4] Nens                   Dona sense estat civil Home clase alta
## [7] Sacerdot               Doctor
## 9 Levels: Doctor Dona casada Dona clase alta ... Sacerdot
```

Apliquem el model de randomforest sobre le subdataset on està informat si sobreviu el passatger

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(0)

rf<-randomForest(Survived~Sex + Cabin +Embarked+Title_refactor+Age+classe+SibSp+Parch+Fare,data=train_cl
importance(rf)
```

```
##                 MeanDecreaseGini
## Sex                      54.10121
## Cabin                    26.31377
## Embarked                 11.08166
## Title_refactor           76.93667
## Age                      63.12208
## classe                   29.33415
## SibSp                    20.65134
## Parch                    10.12543
## Fare                     64.03164
```

```
prediction<-predict(rf,newdata=test_clean)

PassengerId<-test_clean$PassengerId
output.df<-as.data.frame(PassengerId)
output.df$Survived<-prediction

write.csv(output.df,file="/Users/PcCom/Desktop/UOC/Tipologia i cicle de dades/kaggle_submission_1.csv",
```

Realitzem una prova utilitzant regresió logística

```
model<-glm(Survived~.,family=binomial(link='logit'),data=train_clean)
```

Finalment, guardem el resultat de la predicció del subdataset on no està informat si el supervivent sobreviu.

```
prediction<-predict(model,newdata=test_clean)
prediction_r<-ifelse(prediction>0.5,1,0)
PassengerId<-test_clean$PassengerId
output.df<-as.data.frame(PassengerId)
output.df$Survived<-prediction_r

write.csv(output.df,file="/Users/PcCom/Desktop/UOC/Tipologia i cicle de dades/kaggle_submission_3.csv",
```

Realitzem prova amb arbres de decisió

```
library(rpart)
model_tree<-rpart(Survived~.,data=train_clean,method = 'class')
```

```
prediction<-predict(model_tree,newdata=test_clean,type = 'class')

PassengerId<-test_clean$PassengerId
output.df<-as.data.frame(PassengerId)
output.df$Survived<-prediction

write.csv(output.df,file="/Users/PcCom/Desktop/UOC/Tipologia i cicle de dades/kaggle_submission_4.csv",
```

0.4.4 Resultat

Al pujar el resultat a Kaggle s'ha obtingut un resultat de 77.99% el qual és un resultat bo, millorable.

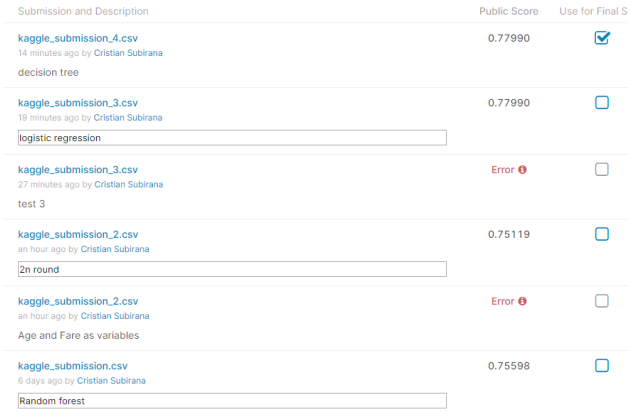


Figure 3: Score Kaggle