

Pràctica Final

Cristian Subirana

25 de Decembre de 2019

Contents

0.1	Descripció del dataset	2
0.1.1	Presentació de les dades	2
0.2	Integració i selecció de les dades	2
0.3	Neteja de les dades	6
0.3.1	Variables PassengerId i Survived	7
0.3.2	Variable Name	7
0.3.3	Variable Age	9
0.3.4	Variable Embarked	12
0.3.5	Variable Cabin	12
0.3.6	Variable Pclass	13
0.3.7	Variable SibSp	13
0.3.8	Variable Parch	14
0.3.9	Variable Fare	14
0.3.10	Dataset Final	15
0.4	Anàlisi de les dades	17
0.4.1	Test de normalitat	17
0.4.2	Distribució de les variables segons si sobreviu	19
0.4.2.1	Factor sobreviu segons classe	19
0.4.2.2	Factor sobreviu segons Port d'embarcació	19
0.4.2.3	Factor sobreviu segons Títol	20
0.4.2.4	Factor sobreviu segons Rang d'edat	21
0.4.2.5	Factor sobreviu segons Rand de valor de ticket	22
0.4.2.6	Factor sobreviu segons Si té parella	23
0.4.2.7	Factor sobreviu segons Si té Pares o fills a bord	24
0.4.2.8	Factor sobreviu segons Sexe	25
0.4.3	Model de regressió per a poder predir supervivents	28
0.4.4	Resultat	29

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 train.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 evaluarem 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 majoritàriament eren persones que viatjaven amb classe mitja, amb una mitjana d'edat d'uns 30 anys, dels quals majoritàriament viatjaven sols.

Cal considerar que a la variable Age, Fare són les úniques variables que contenen NA's pel que en el següent apartat 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 anàlisi 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.

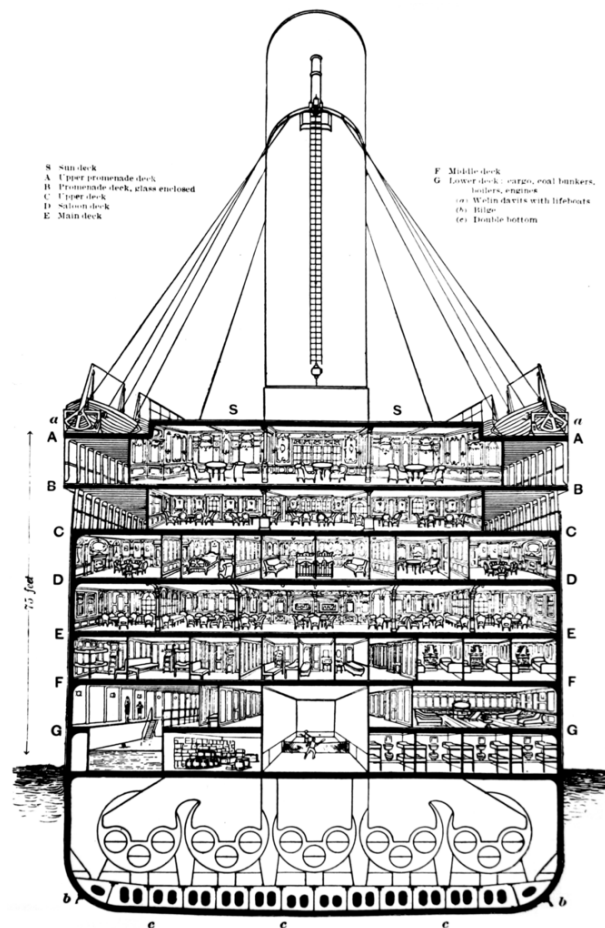


Figure 1: Titanic

Observe'm en la imatge, que a major lletra, més aprop del borts 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 dels 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 preparen 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 títol.

Separare'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 títols 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, independentment del seu estat civil Sir->Home de classe alta o distinguit per la seva professió o conducta, respecte o cortesia Mme->Dona de classe alta o distinguida per la seva professió o conducta, respecte

o cortesia, equivaldria a Madam Sir the Countess->Home amb carrec molt important, compte Lady->Forma aducada de dirigir-se a una dona Capt->Capità, mariner com a professió Major->carrec molt important dins d'una area Master-> Per referir-se a nens Rev->Sacerdot Dr-> Doctor en medicina Mlle->Equivalent de mademoiselle o de Miss Jonkheer->títol nobiliari específic de la família 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 títols, 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 viatge 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","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>
```

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.

Crec que la opció 2 seria la més idonea, ja que en la opció 1, pot apareixer un important esbiaix.

```
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
```

```
##NENS
#Selecionem aquells que tinguin NA per cada categoria
nens_q<-train_c1[train_c1$Title_refactor=="Nens",]
nens_z<-nens_q[!is.na(nens_q$Age),]
nens_var<-mean(nens_z$Age)
nens_refactor<-filter(train_c1,(train_c1$Title_refactor=="Nens" & is.na(train_c1$Age)))

#On hi haguin NA, posari la mitjana del grup
nens_refactor$Age<-nens_var
train_c1$Age[train_c1$Title_refactor=="Nens" & is.na(train_c1$Age)]<-nens_var

##DONA SOLTERA
#Selecionem aquells que tinguin NA per cada categoria
dona_soltera_q<-train_c1[train_c1$Title_refactor=="Dona soltera",]
dona_soltera_z<-dona_soltera_q[!is.na(dona_soltera_q$Age),]
dona_soltera_var<-mean(dona_soltera_z$Age)
```

```

dona_soltera_refactor<-filter(train_c1,(train_c1$Title_refactor=="Dona soltera" & is.na(train_c1$Age)))

#On hi haguin NA, posari la mitjana del grup
dona_soltera_refactor$Age<-dona_soltera_var
train_c1$Age[train_c1$Title_refactor=="Dona soltera" & is.na(train_c1$Age)]<-dona_soltera_var

##DONA CASADA
#Seleccióem aquells que tinguin NA per cada categoria
dona_casada_q<-train_c1[train_c1$Title_refactor=="Dona casada",]
dona_casada_z<-dona_casada_q[!is.na(dona_casada_q$Age),]
dona_casada_var<-mean(dona_casada_z$Age)
dona_casada_refactor<-filter(train_c1,(train_c1$Title_refactor=="Dona casada" & is.na(train_c1$Age)))

#On hi haguin NA, posari la mitjana del grup
dona_casada_refactor$Age<-dona_casada_var
train_c1$Age[train_c1$Title_refactor=="Dona casada" & is.na(train_c1$Age)]<-dona_casada_var

##Home sense estat civil
#Seleccióem aquells que tinguin NA per cada categoria
home_sns_civil_q<-train_c1[train_c1$Title_refactor=="Home sense estat civil",]
home_sns_civil_z<-home_sns_civil_q[!is.na(home_sns_civil_q$Age),]
home_sns_civil_var<-mean(home_sns_civil_z$Age)
home_sns_civil_refactor<-filter(train_c1,(train_c1$Title_refactor=="Home sense estat civil" & is.na(train_c1$Age)))

#On hi haguin NA, posari la mitjana del grup
home_sns_civil_refactor$Age<-home_sns_civil_var
train_c1$Age[train_c1$Title_refactor=="Home sense estat civil" & is.na(train_c1$Age)]<-home_sns_civil_var

##Doctor
#Seleccióem aquells que tinguin NA per cada categoria
doctor_q<-train_c1[train_c1$Title_refactor=="Doctor",]
doctor_z<-doctor_q[!is.na(doctor_q$Age),]
doctor_var<-mean(doctor_z$Age)
doctor_refactor<-filter(train_c1,(train_c1$Title_refactor=="Doctor" & is.na(train_c1$Age)))

#On hi haguin NA, posari la mitjana del grup
doctor_refactor$Age<-doctor_var
train_c1$Age[train_c1$Title_refactor=="Doctor" & is.na(train_c1$Age)]<-doctor_var

##Dona sense estat civil
#Seleccióem aquells que tinguin NA per cada categoria
dona_snse_civil_q<-train_c1[train_c1$Title_refactor=="Dona sense estat civil",]
dona_snse_civil_z<-dona_snse_civil_q[!is.na(dona_snse_civil_q$Age),]
dona_snse_civil_var<-mean(dona_snse_civil_z$Age)
dona_snse_civil_refactor<-filter(train_c1,(train_c1$Title_refactor=="Dona sense estat civil" & is.na(train_c1$Age)))

#On hi haguin NA, posari la mitjana del grup
dona_snse_civil_refactor$Age<-dona_snse_civil_var
train_c1$Age[train_c1$Title_refactor=="Dona sense estat civil" & is.na(train_c1$Age)]<-dona_snse_civil_var

```

Els camps Sacerdot, Dona classe alta i Home classe alta no contienien NA's pel que s'ha omés la seva part de codi. Comprove'm el resultat, en que no apareixen més NA's al camp Age

```
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.:21.80   1st Qu.:0.0000   1st Qu.:0.000
## Mode  :character Median :30.00   Median :0.0000   Median :0.000
##                      Mean  :29.91   Mean  :0.4989   Mean  :0.385
##                      3rd Qu.:36.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                      Max.   :80.00   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
## Length:1309     Length:1309
## Class :character  Class :character
## Mode  :character  Mode  :character
##
##
##
##
```

Finalment veiem que ja no hi han NA's.

Una vegada ja el camp Age ja no conté NA's, definim rangs d'edat per als dos entorns.

Primerament creem la variable

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

Definim rangs:

```
train_c1$age_rang[round(train_c1$Age,digits=0)>=0 & train_c1$Age<=10]<-"0-10"
train_c1$age_rang[round(train_c1$Age,digits=0)>=11 & round(train_c1$Age,digits=0)<=18]<-"11-18"
train_c1$age_rang[round(train_c1$Age,digits=0)>=19 & round(train_c1$Age,digits=0)<=25]<-"19-25"
train_c1$age_rang[round(train_c1$Age,digits=0)>=26 & round(train_c1$Age,digits=0)<=35]<-"26-35"
train_c1$age_rang[round(train_c1$Age,digits=0)>=36 & round(train_c1$Age,digits=0)<=45]<-"36-45"
```

```
train_c1$age_rang[round(train_c1$Age,digits=0)>=46 & round(train_c1$Age,digits=0)<=60]<-"46-60"
train_c1$age_rang[round(train_c1$Age,digits=0)>=61 & round(train_c1$Age,digits=0)<=80]<-"61-80"
train_c1$age_rang[round(train_c1$Age,digits=0)>80]<-">80"
```

Comprove'm resultat

```
unique(train_c1$age_rang)
```

```
## [1] "19-25" "36-45" "26-35" "46-60" "0-10" "11-18" "61-80"
```

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 referència 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 distribució 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 accessibles 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

Creem una nova variable al dataset

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

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

```
train_c1$TeParella[train_c1$SibSp>0]<-'Si'  
train_c1$TeParella[train_c1$SibSp==0]<-'No'
```

Comprove'm resultat

```
unique(train_c1$TeParella)
```

```
## [1] "Si" "No"
```

0.3.8 Variable Parch

Creem una nova variable al dataset

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

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

```
train_c1$TeParesOFillsAbord[train_c1$Parch>0]<-'Si'  
train_c1$TeParesOFillsAbord[train_c1$Parch==0]<-'No'
```

Comprove'm resultat

```
unique(train_c1$TeParesOFillsAbord)
```

```
## [1] "No" "Si"
```

0.3.9 Variable Fare

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

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

```
train_c1$Fare_rang[ceiling(train_c1$Fare)<=5]<-"0-5"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=6 & ceiling(train_c1$Fare)<=10]<-"6-10"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=11 & ceiling(train_c1$Fare)<=15]<-"11-15"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=16 & ceiling(train_c1$Fare)<=20]<-"16-20"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=21 & ceiling(train_c1$Fare)<=30]<-"21-30"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=31 & ceiling(train_c1$Fare)<=50]<-"31-50"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=51 & ceiling(train_c1$Fare)<=100]<-"51-100"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=101 & ceiling(train_c1$Fare)<=150]<-"101-150"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=151 & ceiling(train_c1$Fare)<=200]<-"151-200"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=201 & ceiling(train_c1$Fare)<=300]<-"201-300"  
train_c1$Fare_rang[ceiling(train_c1$Fare)>=301]<-">301"
```

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”.

```
unique(train_c1$Fare_rang)
```

```
## [1] "6-10" "51-100" "21-30" "11-15" "31-50" "16-20" "201-300"
## [8] "101-150" "0-5" ">301" "151-200" ""
```

Ens trobe'm que a test hi ha un passatger que no té valor de ticket, el qual és NA. Per aquest motiu haure'm de substituir el valor.

```
train_c1[train_c1$Fare_rang=="",]
```

```
## # A tibble: 1 x 18
## PassengerId Survived Pclass Name Sex Age SibSp Parch Fare Cabin
## <int> <int> <int> <chr> <chr> <dbl> <int> <int> <dbl> <chr>
## 1 1044 NA 3 Stor~ male 61 0 0 NA Z
## # ... with 8 more variables: Embarked <chr>, Title <chr>,
## # Title_refactor <chr>, age_rang <chr>, classe <chr>, TeParella <chr>,
## # TeParesOFillsAbord <chr>, Fare_rang <chr>
```

Calcule'm la mitjana de passatgers amb atributs semblants al nostre cas i substituïm el NA per ell

```
mean(train_c1$Fare[!is.na(train_c1$Fare) & train_c1$Pclass==3 & train_c1$Embarked=="S" & train_c1$Title,
```

```
## [1] 11.73956
```

```
train_c1$Fare_rang[train_c1$Fare_rang==""]<-11.42149
```

Comprove'm que ja no hi ha cap NA

```
train_c1[train_c1$Fare_rang=="",]
```

```
## # A tibble: 0 x 18
## # ... with 18 variables: PassengerId <int>, Survived <int>, Pclass <int>,
## # Name <chr>, Sex <chr>, Age <dbl>, SibSp <int>, Parch <int>,
## # Fare <dbl>, Cabin <chr>, Embarked <chr>, Title <chr>,
## # Title_refactor <chr>, age_rang <chr>, classe <chr>, TeParella <chr>,
## # TeParesOFillsAbord <chr>, Fare_rang <chr>
```

0.3.10 Dataset Final

Tenir en compte que alhora de preparar i netejar les variables s'ha fet ús de “feature engineering” a fi de millorar el rendiment del model, ja que es pretén millorar les variables per a que acabin resultant en millor resultats.

Prepare'm el nou dataset:

```
train_net<-train_c1[,-which(names(train_c1) %in% c("Pclass","Name","Age","SibSp","Parch","Fare","Title"))]
```

```
#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$age_rang<-as.factor(train_net$age_rang)
train_net$classe<-as.factor(train_net$classe)
train_net$TeParella<-as.factor(train_net$TeParella)
train_net$TeParesOFillsAbord<-as.factor(train_net$TeParesOFillsAbord)
train_net$Fare_rang<-as.factor(train_net$Fare_rang)
train_net$Survived<-as.factor(train_net$Survived)
```

Degut a que hàviem fusionat test i train, tenim NA's a la variable Survived, pel que si volem veure la distribució de supervivents per cada variable, els ometre'm.

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

Nou dataset de train:

```
summary(train_net)
```

```
## PassengerId Survived Sex Cabin Embarked
## Min. : 1.0 0:549 female:314 Z :687 C:168
## 1st Qu.:223.5 1:342 male :577 C : 59 Q: 77
## Median :446.0 B : 47 S:646
## Mean :446.0 D : 33
## 3rd Qu.:668.5 E : 32
## Max. :891.0 A : 15
## (Other): 18
## Title_refactor age_rang classe TeParella
## Home sense estat civil:518 0-10 : 68 Alta :216 No:608
## Dona soltera :184 11-18: 75 Baixa:491 Si:283
## Dona casada :125 19-25:198 Mitja:184
## Nens : 40 26-35:315
## Home classe alta : 8 36-45:132
## Doctor : 7 46-60: 81
## (Other) : 9 61-80: 22
## TeParesOFillsAbord Fare_rang
## No:678 6-10 :319
## Si:213 21-30 :142
## 11-15 :122
## 51-100 :107
## 31-50 : 74
## 16-20 : 57
## (Other): 70
```


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 numèriques, 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 numèriques.

```
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.:21.80      1st Qu.:0.0000      1st Qu.:0.000
## Mode  :character  Median :30.00      Median :0.0000      Median :0.000
##                Mean   :29.91      Mean   :0.4989      Mean   :0.385
##                3rd Qu.:36.00      3rd Qu.:1.0000      3rd Qu.:0.000
##                Max.   :80.00      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_rang
## Length:1309      Length:1309      Length:1309
## Class :character      Class :character      Class :character
## Mode  :character      Mode  :character      Mode  :character
##
##
##
##
##      classe      TeParella      TeParesOFillsAbord
## Length:1309      Length:1309      Length:1309
## Class :character      Class :character      Class :character
## Mode  :character      Mode  :character      Mode  :character
##
##
##
##      Fare_rang
## Length:1309
## Class :character
## Mode  :character
```

```
##  
##  
##  
##
```

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

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  train_c1$Pclass  
## W = 0.72337, p-value < 2.2e-16
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  train_c1$Age  
## W = 0.97495, p-value = 2.678e-14
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  train_c1$SibSp  
## W = 0.51108, p-value < 2.2e-16
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  train_c1$Parch  
## W = 0.49797, p-value < 2.2e-16
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  train_c1$Fare  
## W = 0.52782, p-value < 2.2e-16
```

El resultat del test indica que cap de les variables numèriques 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.

0.4.2 Distribució de les variables segons si sobreviu

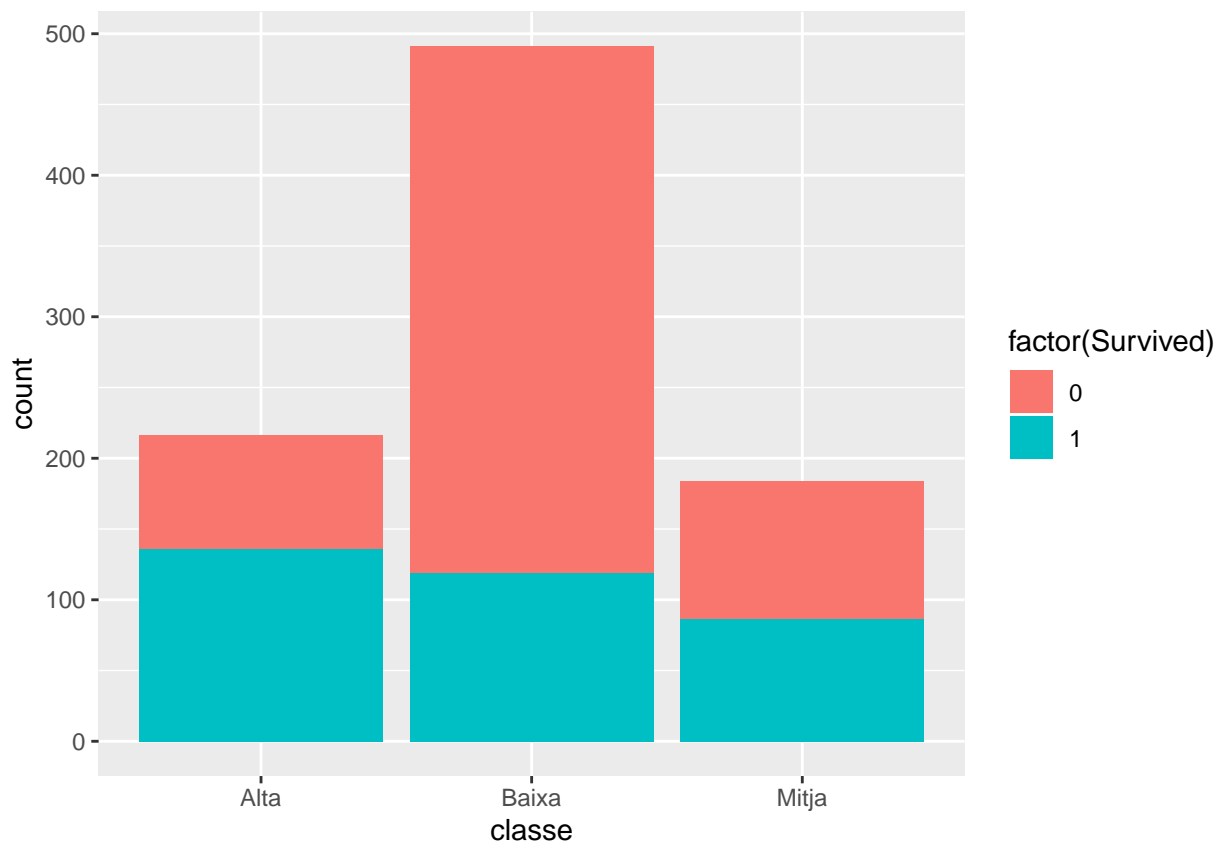
A continuació volem veure per cada variable, els volums de passatgers segons si sobreviu o no, a fi de tenir una visió general de l'importància de cada variable.

0.4.2.1 Factor sobreviu segons classe

```
library(ggplot2)

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

Warning: Ignoring unknown parameters: position

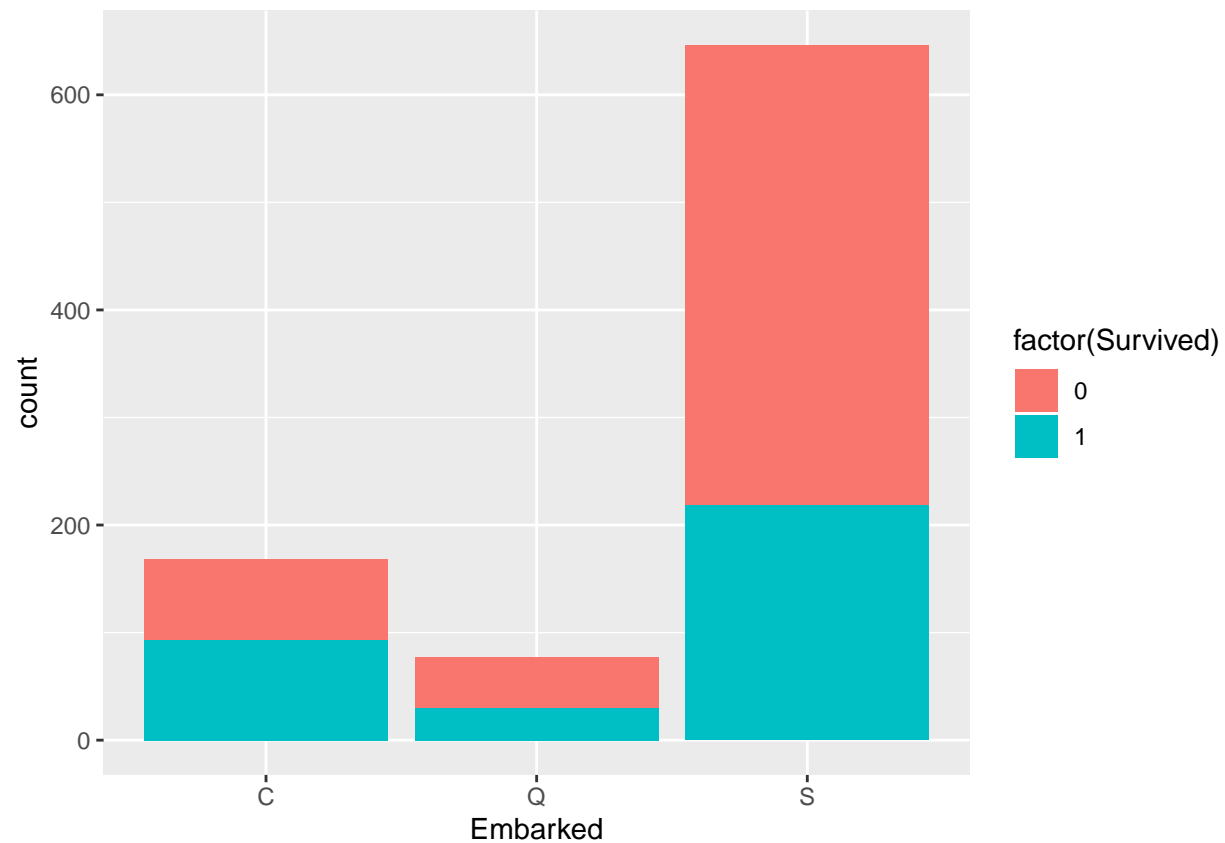


0.4.2.2 Factor sobreviu segons Port d'embarcació

```
library(ggplot2)

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

Warning: Ignoring unknown parameters: position

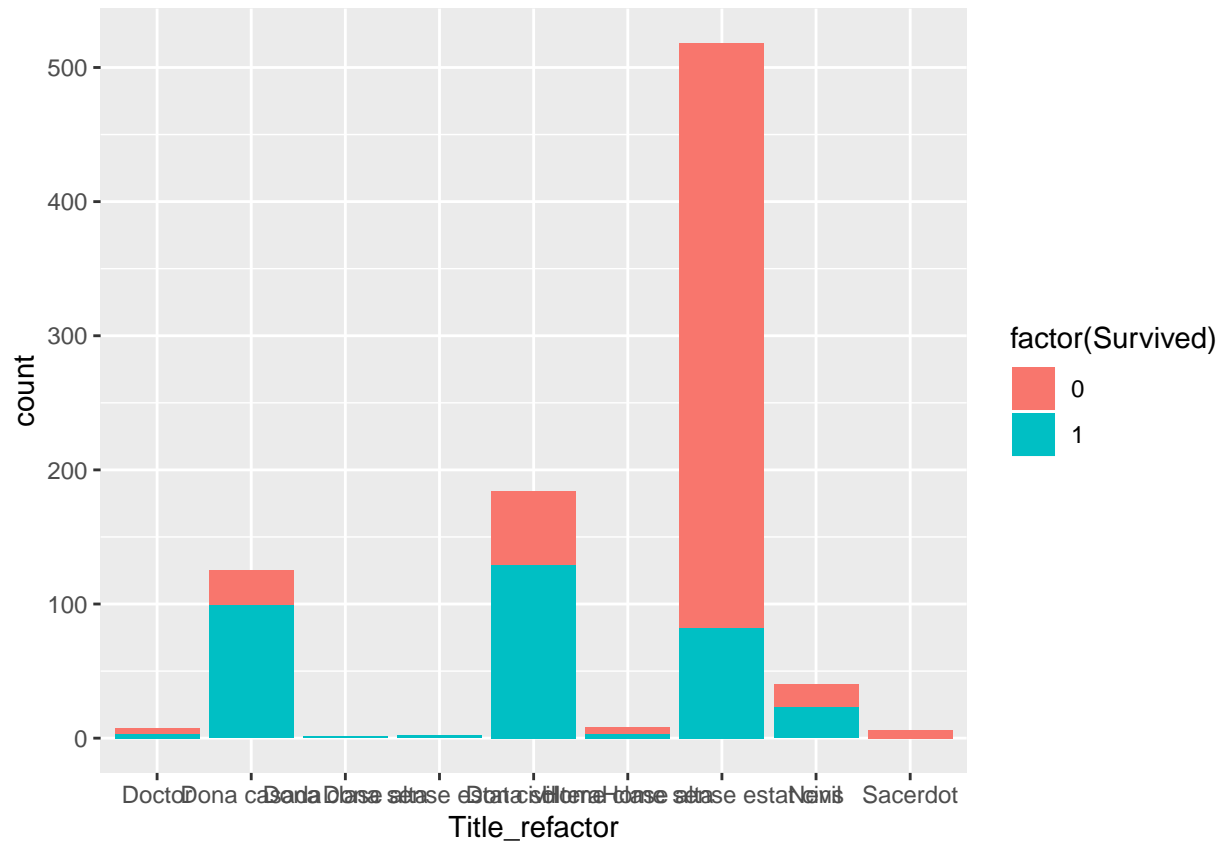


0.4.2.3 Factor sobrevisu segons Títol

```
library(ggplot2)

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

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



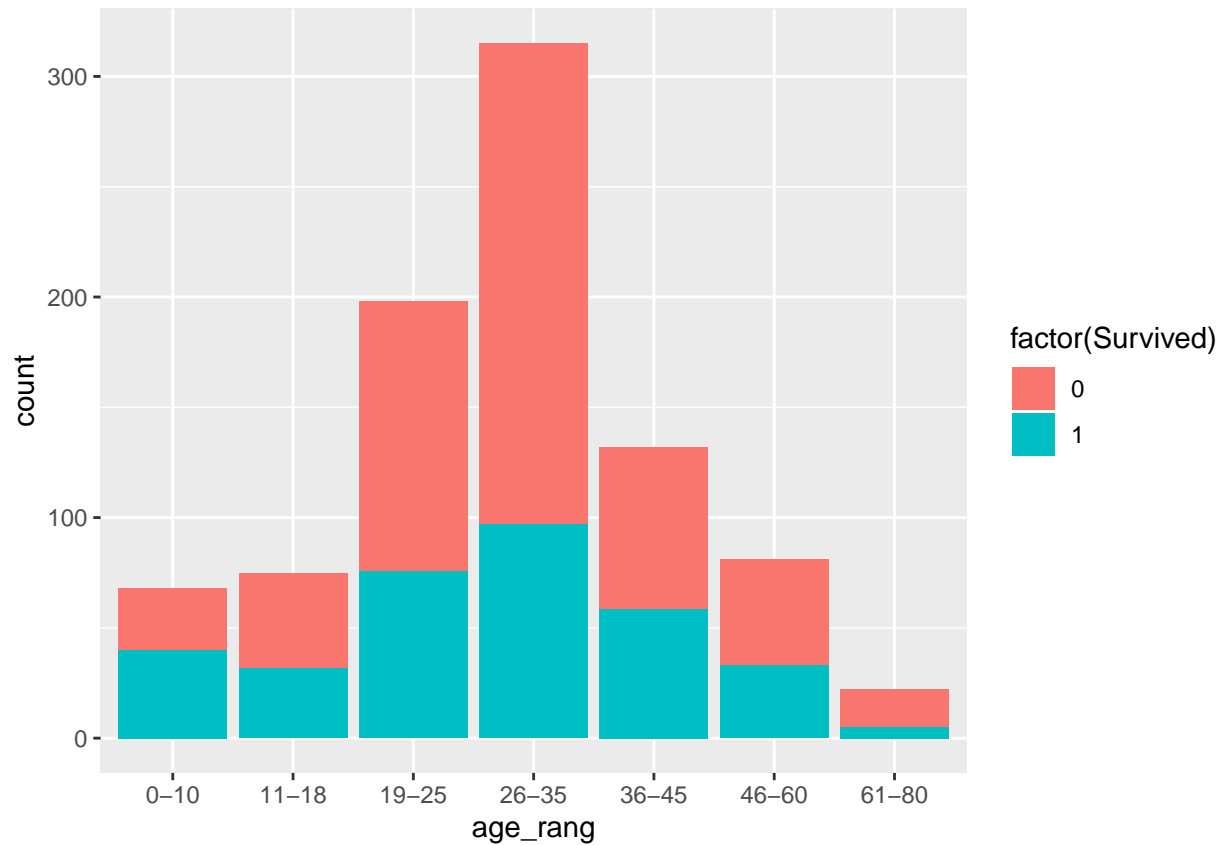
Per a veure correctament aquest gràfic es requereix de mostrar en una nova finestra a pantalla completa. Podem apreciar que ser “Home sense estat civil” pot ser un clar factor de no sobreviure.

0.4.2.4 Factor sobrevisu segons Rang d'edat

```
library(ggplot2)

train_net %>% ggplot(aes(x=age_rang, fill=factor(Survived))) + geom_bar(stat="count", position="fill")

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

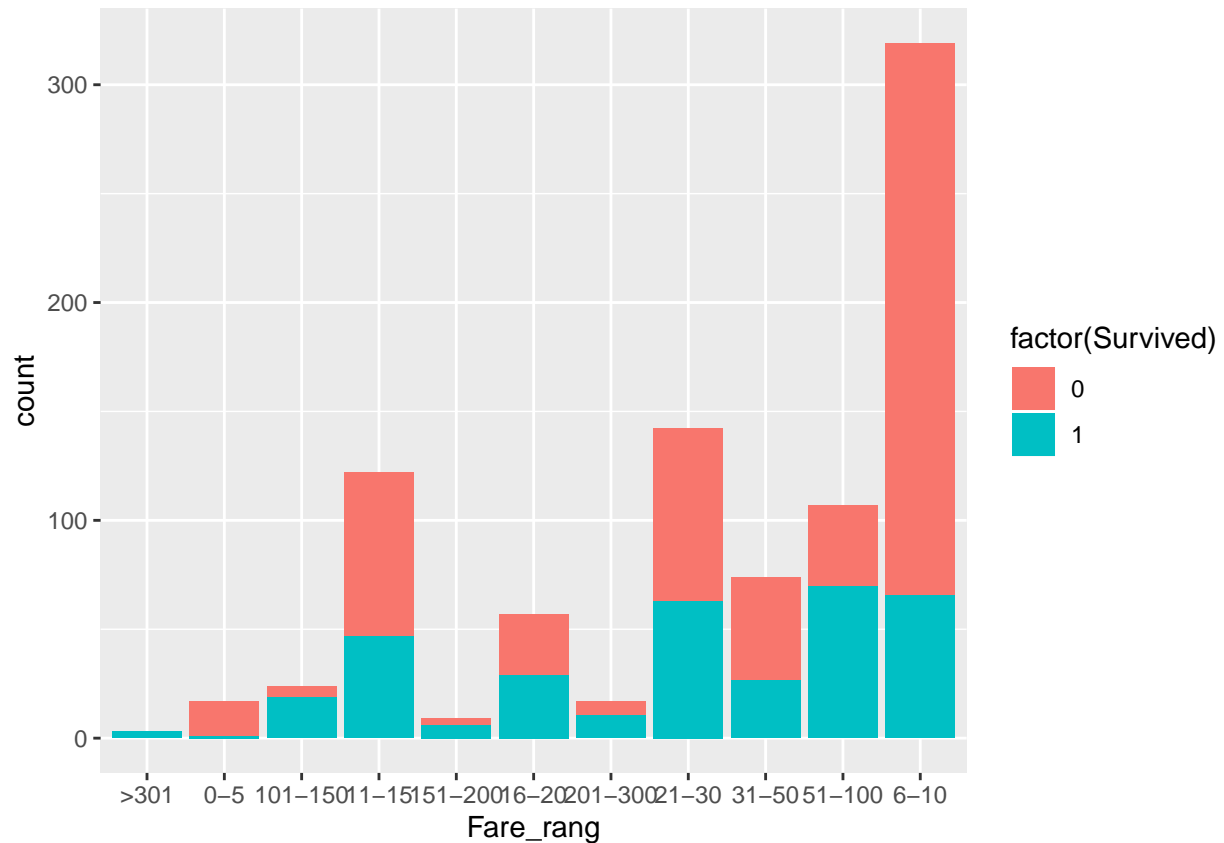


0.4.2.5 Factor sobrevis segons Rang de valor de ticket

```
library(ggplot2)

train_net %>% ggplot(aes(x=Fare_rang, fill=factor(Survived))) + geom_bar(stat="count", position="fill")

## Warning: Ignoring unknown parameters: position
```



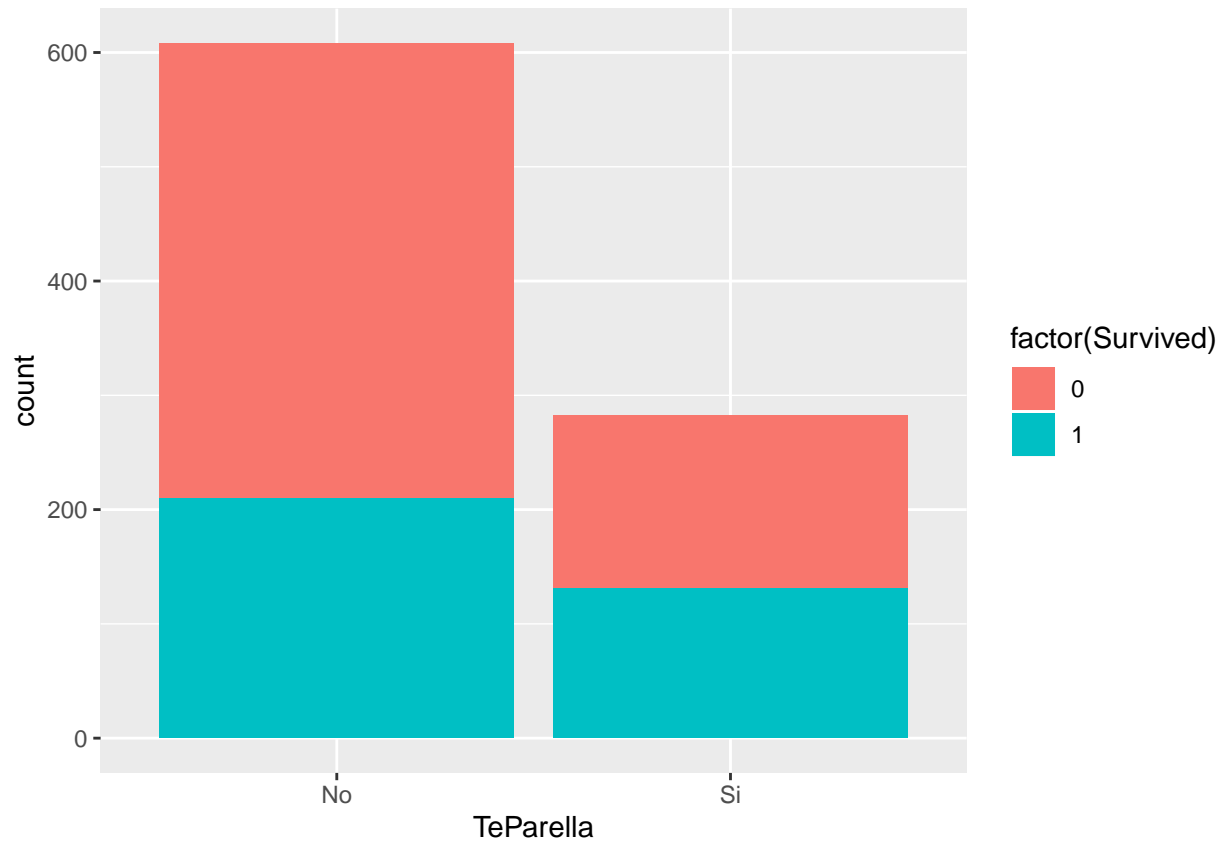
Per apreciar aquest gràfic també és necessari fer ús de una nova finestra en pantalla completa. Observe'm que el valor del ticket elevat és un clar factor per sobreviure.

0.4.2.6 Factor sobreviu segons Si té parella

```
library(ggplot2)

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

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

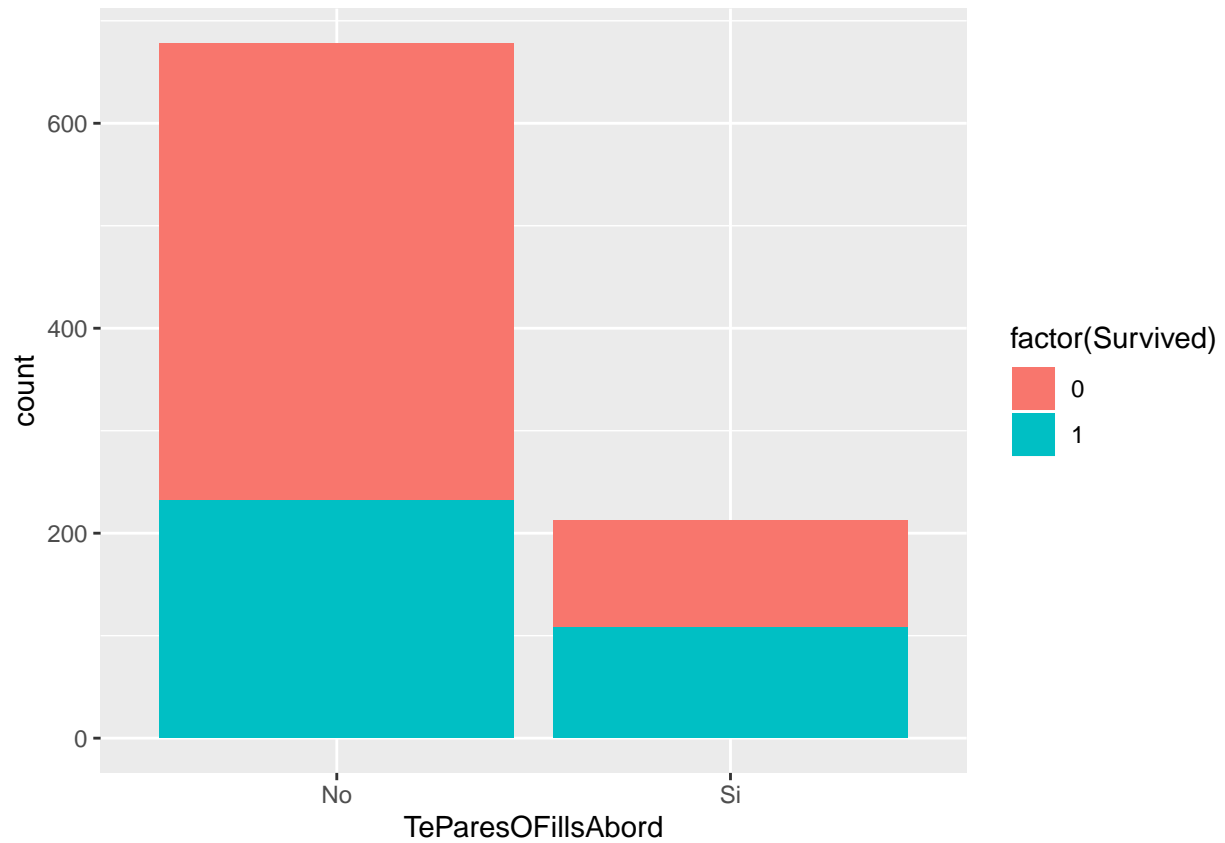


0.4.2.7 Factor sobrevis segons Si té Pares o fills a bord

```
library(ggplot2)
```

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

```
## Warning: Ignoring unknown parameters: position
```

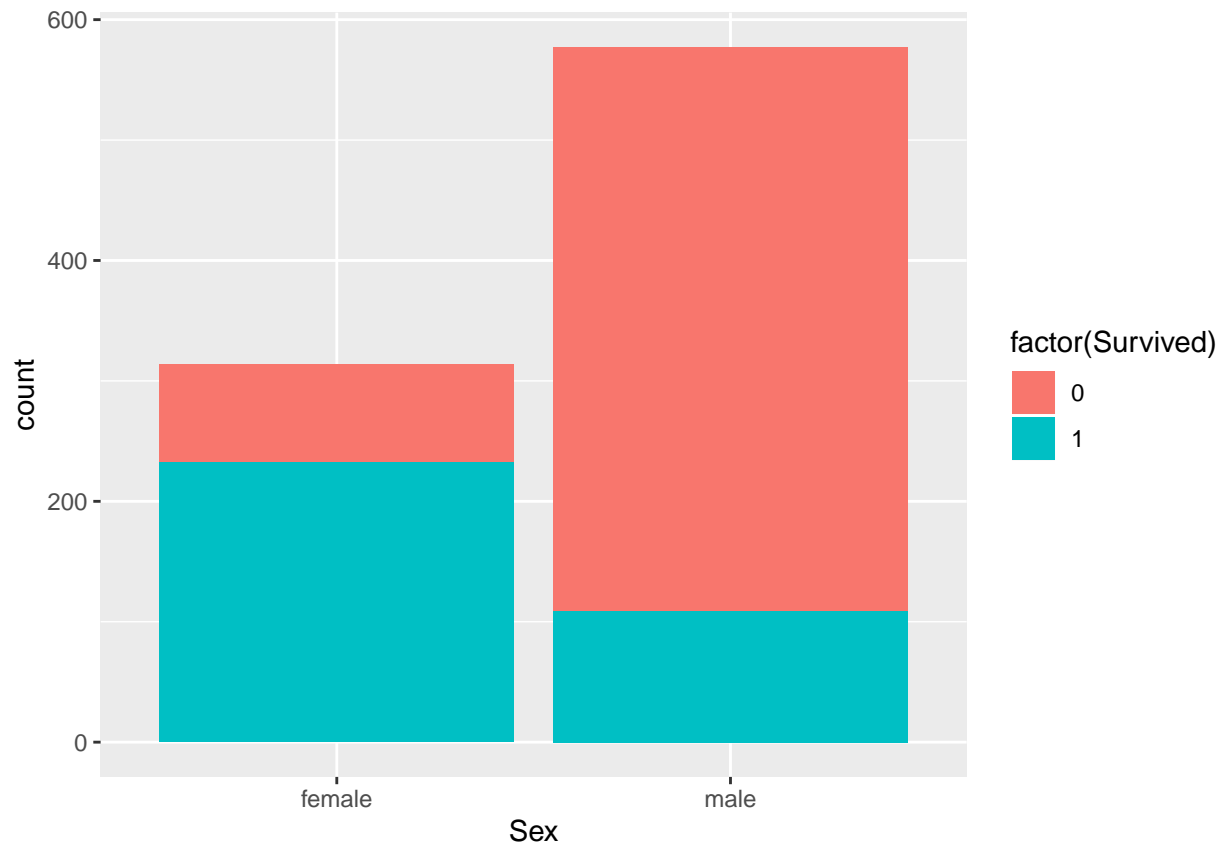



0.4.2.8 Factor sobrevisu segons Sexe

```
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$Survived,train_net$age_rang)

##
## Pearson's Chi-squared test
##
## data: train_net$Survived and train_net$age_rang
## X-squared = 24.962, df = 6, p-value = 0.0003471

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

chisq.test(train_net$Survived,train_net$TeParella)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: train_net$Survived and train_net$TeParella
## X-squared = 11.456, df = 1, p-value = 0.0007128

chisq.test(train_net$Survived,train_net$TeParesOFillsAbord)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: train_net$Survived and train_net$TeParesOFillsAbord
## X-squared = 18.656, df = 1, p-value = 1.565e-05
```

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

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

```
##
## Pearson's Chi-squared test
##
## data: train_net$Survived and train_net$Fare_rang
## X-squared = 118.64, df = 10, 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 part, 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

```
summary(train_clean)
```

```
## PassengerId Survived Sex Cabin Embarked
## Min. : 1.0 0:549 female:314 Z :687 C:168
## 1st Qu.:223.5 1:342 male :577 C : 59 Q: 77
## Median :446.0 B : 47 S:646
## Mean :446.0 D : 33
## 3rd Qu.:668.5 E : 32
## Max. :891.0 A : 15
## (Other): 18
## Title_refactor age_rang classe TeParella
## Home sense estat civil:518 0-10 : 68 Alta :216 No:608
## Dona soltera :184 11-18: 75 Baixa:491 Si:283
## Dona casada :125 19-25:198 Mitja:184
## Nens : 40 26-35:315
## Home classe alta : 8 36-45:132
## Doctor : 7 46-60: 81
## (Other) : 9 61-80: 22
## TeParesOFillsAbord Fare_rang
## No:678 6-10 :319
## Si:213 21-30 :142
## 11-15 :122
## 51-100 :107
## 31-50 : 74
## 16-20 : 57
## (Other): 70
```

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

```
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

## The following object is masked from 'package:dplyr':
##
##     combine

set.seed(0)

rf<-randomForest(Survived~Sex + Cabin +Embarked+Title_refactor+age_rang+classe+TeParella+TeParesOFillsA
importance(rf)

##              MeanDecreaseGini
## Sex              52.769261
## Cabin            28.142808
## Embarked         14.312295
## Title_refactor   80.339492
## age_rang         28.698809
## classe          33.772513
## TeParella        9.785946
## TeParesOFillsAbord 6.837893
## Fare_rang       39.531454
```

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

```
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.csv",row.names=FALSE)
```


0.4.4 Resultat

Al pujar el resultat a Kaggle s'ha obtingut un resultat de 75.55% el qual és un resultat bo, millorable.

Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

 Kaggle · 16,543 teams · Ongoing

OverviewDataNotebooksDiscussionLeaderboardRulesTeamMy SubmissionsSubmit Predictions

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
kaggle_submission.csv	7 minutes ago	1 seconds	0 seconds	0.75598

Complete

[Jump to your position on the leaderboard](#)

Make a submission for [Cristian Subirana](#)

You have 9 submissions remaining today. This resets 7 hours from now (00:00 UTC).

Figure 3: Score Kaggle