R Notebook

Xavier Parramon Boada

1. Detalls de l'activitat

1.1. Descripció

En aquesta pràctica s'elabora un cas pràctic orientat a aprendre a identificar les dades rellevants per a un projecte analític i usar les eines d'integració, neteja, validació i anàlisi de les mateixes.

1.2. Objectius

Els objectius concrets d'aquesta pràctica són:

- Aprendre a aplicar els coneixements adquirits i la seva capacitat de resolució de problemes en entorns nous o poc coneguts dintre de contextos més amplis o multidisciplinaris.
- Saber identificar les dades rellevants i els tractaments necessaris)Integració, neteja i validació) per dur a terme un projecte analític.
- Aprendre a analitzar les dades adequadament per abordar la informació continguda en les dades.
- identificar la millor representació dels resultats per tal d'aportar conclusions sobre el problema plantejat en el procés analític.
- Actuar amb els principis ètics i legals relacionats amb la manipulació de dades en funció de l'àmbit d'aplicació.
- Desenvolupar les habilitats d'aprenentatge que els permetin continuar estudiant d'una manera que haurà de ser en gran manera autodirigida o autònoma.
- Desenvolupar la capacitat de cerca, gestió i ús d'informació i recursos en l'àmbit de la ciència de dades.

1.3. Competències

En aquesta pràctica es desenvolupen les següents competències del Màster de Data Science:

- Capacitat d'analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per abordar-lo i resoldre'l.
- Capacitat per aplicar les tècniques específiques de tractament de dades (integració, transformació, neteja i validació) per al seu posterior anàlisi.

2. Resolució

Procedim amb la resolució de la pràctica.

2.1. Descripció del dataset.

El dataset seleccionat s'anomena *"Titanic: Machine Learning from Disaster"*, i s'ha obtingut a partir de l'enllaç de Kaggel. El dataset recull un conjunt d'informació referent als passatgers que viatjaven en el titànic, i l'objectiu d'aquest és crear un model que sigui capaç de predir si els passatgers va sobreviure o no a l'accident a partir de diferents paràmetres. Els atributs que podem trobar son:

- **PassengerId**: identificador del passatger del titànic.
- **Survived**: Supervivencia a l'accident. (0=no, 1= Si)
- **Pclass**: Classe del passatger. (1=1ra,2=2na,3=3ra)
- **Name**: Nom del passatger.
- **Sex**: Sexe del passatger.
- **Age**: Edat del passatger en anys.
- **SibSp**: Nombre de germans/cònjuges a bord del titànic.
- **Parch**: Nombre de pares/fills a bord del titànic.
- **Ticket**: numero del tiquet.
- **Fare**: tarifa del passatger.
- **Cabin**: numero de cabina.
- **Embarked**: Port de l'embarc. (C=Cherbourg, Q=Queenstown, S=Southampton)

Com que l'objectiu és crear un model predictiu disposem de 2 datasets (train.csv amb 891 registres i test.csv amb 418 registres), la diferencia és que el dataset train conte totes les dades disponibles per a crear i entrenar el model i el dataset test les dades necessàries per fer les prediccions, és a dir test conte dades semblants a train excepte de al informació de si va sobre viure o no.

També inclou un altre csv *"gender_submission.csv"* com a exemple del format de l'arxiu resultant que s'ha d'entregar per la competició. Per a la pràctica no es rellevant.

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

El primer que farem és carregar les dades dels diferents datasets, estudiar-les i analitzar-les, amb l'objectiu de descobrir quins atributs ens aporten més informació per a la creació del model predictiu. Comencem carregant les dades dels 2 csv. Una opció per a facilitar la neteja, seria unir els 2 csv en un únic dataset, d'aquesta manera només tindríem que fer el procés de neteja una sola vegada i després el podríem tornar a separar. Però, per a assegurar que no es barrejant les dades i per a simular que obtenim 2 datasets diferents en el temps, un primer per a crear el model i un segon més tard per a utilitzar-lo amb el model, farem la neteja per separat.

```
#Importem els datasets
train <- read.csv("../data/train.csv",header=TRUE)
test <- read.csv("../data/test.csv",header=TRUE)
head(train,5)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
##                                                  Name    Sex Age SibSp Parch
## 1                             Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                              Heikkinen, Miss. Laina female  26     0     0
```

```
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                             Allen, Mr. William Henry   male  35     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
```

```
head(test,5)
```

```
##   PassengerId Pclass                                          Name    Sex  Age
## 1         892      3                              Kelly, Mr. James   male 34.5
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female 47.0
## 3         894      2                     Myles, Mr. Thomas Francis   male 62.0
## 4         895      3                              Wirz, Mr. Albert   male 27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
##   SibSp Parch  Ticket    Fare Cabin Embarked
## 1     0     0  330911  7.8292              Q
## 2     1     0  363272  7.0000              S
## 3     0     0  240276  9.6875              Q
## 4     0     0  315154  8.6625              S
## 5     1     1 3101298 12.2875              S
```

Les dades dels 2 csv s'han carregat correctament. Seguim amb un anàlisi ràpid del tipus de dades i el rang de valors que poden prendre cada un dels atributs.

```
print("Train:")
```

```
## [1] "Train:"
```

```
str(train)
```

```
## '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       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 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     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
summary(train)
```

```
##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
```

```
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##                                     Name         Sex           Age
##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00
##  Abelson, Mr. Samuel                   :  1                Mean   :29.70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00
##  Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.00
##  (Other)                               :885                NA's   :177
##      SibSp           Parch             Ticket         Fare
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33
##                                   (Other) :852
##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
##  (Other)    :186
```

```
print("Tests:")
```

```
## [1] "Tests:"
```

```
str(test)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182 370 85 5
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101 270 ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```
summary(test)
```

```
##   PassengerId         Pclass
##  Min.   : 892.0   Min.   :1.000
##  1st Qu.: 996.2   1st Qu.:1.000
##  Median :1100.5   Median :3.000
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##                                          Name         Sex           Age
##  Abbott, Master. Eugene Joseph             :  1   female:152   Min.   : 0.17
##  Abelseth, Miss. Karen Marie               :  1   male  :266   1st Qu.:21.00
##  Abelseth, Mr. Olaus Jorgensen             :  1                Median :27.00
##  Abrahamsson, Mr. Abraham August Johannes  :  1                Mean   :30.27
##  Abrahim, Mrs. Joseph (Sophie Halaut Easu):  1                3rd Qu.:39.00
##  Aks, Master. Philip Frank                 :  1                Max.   :76.00
##  (Other)                                   :412                NA's   :86
##      SibSp            Parch            Ticket         Fare
##  Min.   :0.0000   Min.   :0.0000   PC 17608:  5   Min.   :  0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   113503  :  4   1st Qu.:  7.896
##  Median :0.0000   Median :0.0000   CA. 2343:  4   Median : 14.454
##  Mean   :0.4474   Mean   :0.3923   16966   :  3   Mean   : 35.627
##  3rd Qu.:1.0000   3rd Qu.:0.0000   220845  :  3   3rd Qu.: 31.500
##  Max.   :8.0000   Max.   :9.0000   347077  :  3   Max.   :512.329
##                                    (Other) :396   NA's   :1
##              Cabin     Embarked
##                 :327   C:102
##  B57 B59 B63 B66:  3   Q: 46
##  A34            :  2   S:270
##  B45            :  2
##  C101           :  2
##  C116           :  2
##  (Other)        : 80
```

Podem observar com efectivament els tipus d'atributs en els 2 datasets son iguals i amb els mateixos rangs de valors, a excepció de la variable *Survived* que és l'objectiu de la predicció del dataset test. Observant els resultats també poden detectar que alguns dels diferents atributs s'han interpretat com a variables quantitatives, degut a que són valors numèrics, però en realitat són variables qualitatives ja que representen un tipus d'informació que te un rang fix de paràmetres i de poca variació, com poden ser els atributs *Pclass*, que representa la classe del passatger (1,2,3) i *Survived*, que és un booleà que representa si el passatger va sobreviure o no.

```
#Passem les variables quantitatives a qualitatives
train$Survived<-as.factor(train$Survived)
train$Pclass<-as.factor(train$Pclass)
test$Pclass<-as.factor(test$Pclass)
```

2.2.1 Selecció de les dades d'intereés

Tot seguit procedim a seleccionar les dades que ens poden ser interesants per el model. De l'apartat anterior ja hem pogut identificar, una seria d'atributs que per la informació que representen no ens aporten cap tipus d'informació útil per a saber si van sobreviure o no, com són: *PassangerId* (identificador del passatger), *Name* (Nom del passatger), *Tiquet* (tiquet del passatger) i *Faré* (tarifa del tiquet). Per tant aquestes variables les podem eliminar.

```
#Eliminar files
train<-subset(train,select=-c(PassengerId,Name,Ticket,Fare) )
test<-subset(test,select=-c(PassengerId,Name,Ticket,Fare))
```

També podem identificar ràpidament atributs que segurament estan relacionats amb la supervivència o no del passatger, com poden ser: *Pclass* (classe del passatger), *Sex* (sexe) i *Age* (edat). La resta de variables tan pot ser que ens puguin aportar informació útil com no, aquestes són: *sibSp* (nombre de germans/cònjuges a bord), *Parch* (Nombre de pares/fills a bord), *Cabin* (cabina del vaixell) i *Embarked* (Port de l'embarc). Com em dit a priori sembla que la informació que aporten no hagi de ser rellevant per al model, però potser hi ha algun tipus de relació que desconeixem o combinat amb altre informació pots ser útil, per tant la mantindrem. Per exemple, potser els passatgers amb cabines més pròximes als bots salvavides van sobreviure més que els de les cabines més allunyades.

Així, els dataframes resultats són els següents:

```
print("Train:")
```

```
## [1] "Train:"
```

```
str(train)
```

```
## 'data.frame':    891 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 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 ...
##  $ Cabin   : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
print("Test:")
```

```
## [1] "Test:"
```

```
str(test)
```

```
## 'data.frame':    418 obs. of  7 variables:
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age     : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp   : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch   : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Cabin   : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

2.3 Neteja de les dades.

Un cop ja hem carregat les dades i hem fet una primera selecció de les dades d'interès procedim a netejar-les per a eliminar tots els errors presents. La neteja de les dades la realitzem en els 2 datasets per igual, train i test.

2.3.1 Ceros y elements buits

Comencem buscat registres amb valors nuls o perduts. De la anàlisis anterior ja hem pogut detectar que la variable *Age* contenia valors nuls, així que entrem en detall.

```
#Valors Nan i buits
print("train:")
```

```
## [1] "train:"
```

```
print("Nan")
```

```
## [1] "Nan"
```

```
colSums(is.na(train))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch    Cabin Embarked
##        0        0        0      177        0        0        0        0
```

```
print("Buit")
```

```
## [1] "Buit"
```

```
colSums(train=="")
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch    Cabin Embarked
##        0        0        0       NA        0        0      687        2
```

```
print("test:")
```

```
## [1] "test:"
```

```
print("Nan")
```

```
## [1] "Nan"
```

```
colSums(is.na(test))
```

```
##   Pclass      Sex      Age    SibSp    Parch    Cabin Embarked
##        0        0       86        0        0        0        0
```

```
print("Buit")
```

```
## [1] "Buit"
```

```r
colSums(test=="")
```

```
##   Pclass      Sex      Age    SibSp    Parch    Cabin Embarked
##        0        0       NA        0        0      327        0
```

Efectivament veiem que l'atribut *Age* conte dades buides en els 2 datasets (177 a train i 86 a test), també l'atribut *Cabin* conte moltes dades sense valor en els 2 datasets (687 a train i 327 a test) i l'atribut *embarked* conte 2 dades buides en el dataset train. Per a tractar els valors buits o nuls hi ha diferents mètodes, com poden ser eliminar els registres, substituir els valors perduts per una mesura de tendència central, predir o imputar els valors amb mètodes probabilístics o mantenir els valors buit substituint-los per una constant o etiqueta. Anem a veure per a cada cas quina és la millor solució. Comencem per a l'atribut *Embarked*, ja que només hem detectat 2 registres sense valor, podríem optar per eliminar-los, però els substituirem per el valor de tendència per així seguir aprofitant aquestes dades, ja que al ser poques tampoc ens afectaran molt.

```r
#Imputació de valors a Embarked
train[which(train$Embarked==""),"Embarked"]="S"
test[which(test$Embarked==""),"Embarked"]="S"
train$Embarked<-factor(train$Embarked)
test$Embarked<-factor(test$Embarked)
```

El següent atribut es el *Cabin*, la majoria dels seus valors són buit, per tant podríem optar per eliminar directament l'atribut, ja que no sabem si ens aporta o no ens aporta informació, però el que farem és substituir els valors buit per l'etiqueta "No" fent referencia a que el passatge no disposa de cabina, i al mateix temps substituirem la resta de valors per "Si". Per tant el que farem, crear un nou atribut *HasCabin* que pren valors "Si" o "No" i eliminar l'atribut *Cabin*, així corregim els errors a les dades i seguim extreien informació que pot ser útil de l'atribut.

```r
#Nova variable hasCabin
train["HasCabin"]<-ifelse(train$Cabin=="","No","Si")
test["HasCabin"]<-ifelse(test$Cabin=="","No","Si")
#la passem a factor
train$HasCabin<-as.factor(train$HasCabin)
test$HasCabin<-as.factor(test$HasCabin)
#eliminem l'antiga variable
train<-subset(train,select=-Cabin)
test<-subset(test,select=-Cabin)
```

Per últim, queda la variable *Age*, com que el nombre de dades buides es elevat no les podem eliminar, i hi imputarem valors, podríem utilitzar un valor de tendència central, però crec que el més representatiu de la població seria utilitzar un mètode probabilístic per imputar els valors perduts. En aquest cas utilitzarem el mètode del k veïns (*kNN-imputation*), que els que fa es calcular el valor del registre utilitzant els k veïns més pròxims a aquest.

```r
#Importen la llibreria necessaria
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
```

```
#corregim el model amb els 5 veins més propers
train$Age<-kNN(train,k=5)$Age
test$Age<-kNN(test,k=5)$Age
```

Fem la comprovació final de les variables per a comprovar com han quedat, després del tractament de les dades buides o nul · les.

```
#Valors Nan i buits
print("train:")
```

```
## [1] "train:"
```

```
print("Nan")
```

```
## [1] "Nan"
```

```
colSums(is.na(train))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch Embarked HasCabin
##        0        0        0        0        0        0        0        0
```

```
print("Buit")
```

```
## [1] "Buit"
```

```
colSums(train=="")
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch Embarked HasCabin
##        0        0        0        0        0        0        0        0
```

```
summary(train)
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:21.00   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                Median :28.00   Median :0.000   Median :0.0000
##                                Mean   :29.31   Mean   :0.523   Mean   :0.3816
##                                3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  Embarked HasCabin
##  C:168    No:687
##  Q: 77    Si:204
##  S:646
##
##
##
```

```
print("test:")
```

```
## [1] "test:"
```

```
print("Nan")
```

```
## [1] "Nan"
```

```
colSums(is.na(test))
```

```
##   Pclass      Sex      Age    SibSp    Parch Embarked HasCabin
##        0        0        0        0        0        0        0
```

```
print("Buit")
```

```
## [1] "Buit"
```

```
colSums(test=="")
```

```
##   Pclass      Sex      Age    SibSp    Parch Embarked HasCabin
##        0        0        0        0        0        0        0
```

```
summary(test)
```

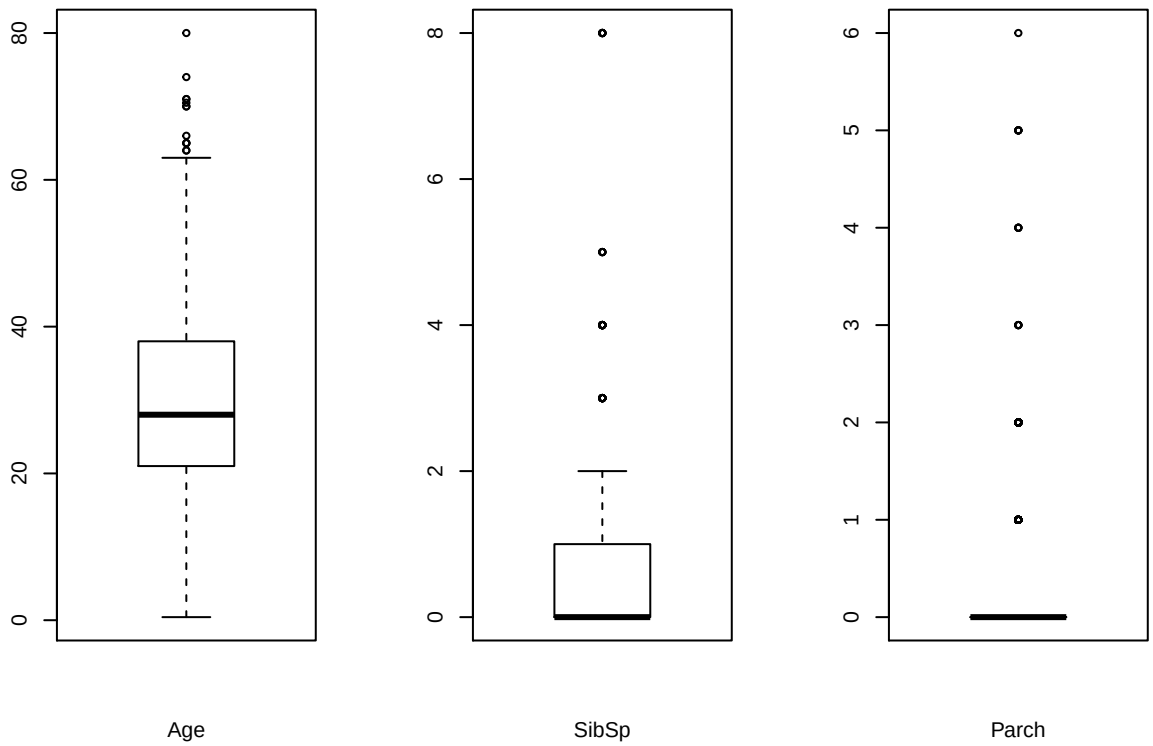
```
##  Pclass      Sex           Age            SibSp           Parch
##  1:107   female:152   Min.   : 0.17   Min.   :0.0000   Min.   :0.0000
##  2: 93   male  :266   1st Qu.:22.00   1st Qu.:0.0000   1st Qu.:0.0000
##  3:218                Median :25.00   Median :0.0000   Median :0.0000
##                       Mean   :29.30   Mean   :0.4474   Mean   :0.3923
##                       3rd Qu.:36.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##                       Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##  Embarked HasCabin
##  C:102    No:327
##  Q: 46    Si: 91
##  S:270
##
##
##
```

Com veiem ja no tenim valors buits ni nuls, i els estadístics de la variable *Age* no han variat gaire respecte als originals al afegir els valors imputats.

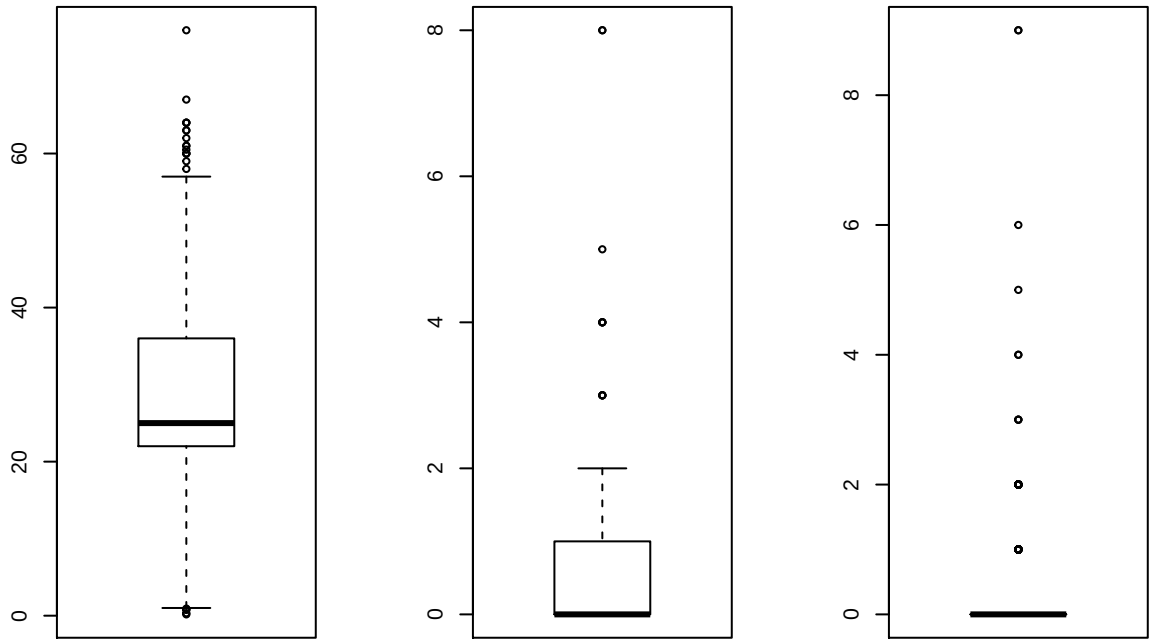
2.3.2 Valors extrems.

Seguim comprovant si hi ha valors extrems o *outliners*. Entenem com a valors extrem aquells valors que es troben molt allunyats de la distribució normal d'una variable o població. Com a criteri general prendrem com a valors extrems tots aquells valors que estan més lluny de 3 desviacions estàndards respecte de la mitjana del conjunt. Per corregí aquets valors, podem eliminar les dades amb valors extrems o substituir aquets valor extrems per el valor més pròxim dins del rang admès, o utilitzar la imputació de valors per mètodes probabilístics. La millor manera de detectar visualment els valors extrems és mitjançant Boxplots. Les variables quantitatives són les úniques que poden tenir valors extrems, ja que les variables qualitatives, tot el seu rang de valors entra dins de la distribució.

```
#Fem els boxplot de les 3 variables quantitatives
par(mfrow=c(1,3))
boxplot(train$Age,xlab="Age")
boxplot(train$SibSp,xlab="SibSp")
boxplot(train$Parch,xlab="Parch")
```




```
par(mfrow=c(1,3))
boxplot(test$Age,xlab="Age")
boxplot(test$SibSp,xlab="SibSp")
boxplot(test$Parch,xlab="Parch")
```



Observem que en totes 3 variables hi ha valors extrems. En el cas de *SibSp* i *Parch*, els valors extrems detectats són els valors que pren la variable exceptuant el valor tendència, això és degut a que la majoria de dades pertanyen al valor tendència, i un nombre molt petit de dades a la resta de valors del rang. Això també ens indica que aquetes dos atributs els podríem haver transformat a dades qualitatives. Decidim mantenir les dades dels valors extrems sense modificació perquè pot ser que ens aportin informació útil.

Respecte a la variable *Age*, veiem que tan tenim valors extrems per sobre com per sota. Observem quin són:

```
print("Train")
```

```
## [1] "Train"
```

```
boxplot.stats(train$Age)$out
```

```
##  [1] 66.0 65.0 71.0 70.5 65.0 64.0 65.0 71.0 64.0 80.0 70.0 70.0 74.0
```

```
print("Test")
```

```
## [1] "Test"
```

```
boxplot.stats(test$Age)$out
```

```
##  [1] 62.00 63.00 60.00 60.00 67.00 76.00 63.00 61.00 60.50 64.00 61.00  0.33
## [13] 60.00 64.00  0.92  0.75 64.00  0.83 58.00  0.17 59.00
```

Veiem que hi ha valors extrems, en el dataset train hi ha 13 dades amb valor igual o superior a 65, al dataset test, hi ha 16 dades amb valor igual o superior a 59 i 4 dades amb valor inferior o igual a 0.92. Tot i això, considero els valors vàlids, ja que el que fa és indicar-nos que hi havia gent gran al vaixell, majors de 59 anys i també nadons de menys d'1 any, no hi ha cap dada que prengui un valor que pugui estar fora del rang d'edat d'una persona, per això deixem els valors extrems tan i com estan.

Exportem els dataset nets

```
#Exportem els datasets
write.csv(train,"../data/train_clean.csv", row.names = TRUE)
write.csv(test,"../data/test_clean.csv", row.names = TRUE)
```

2.4 Anàlisi de les dades

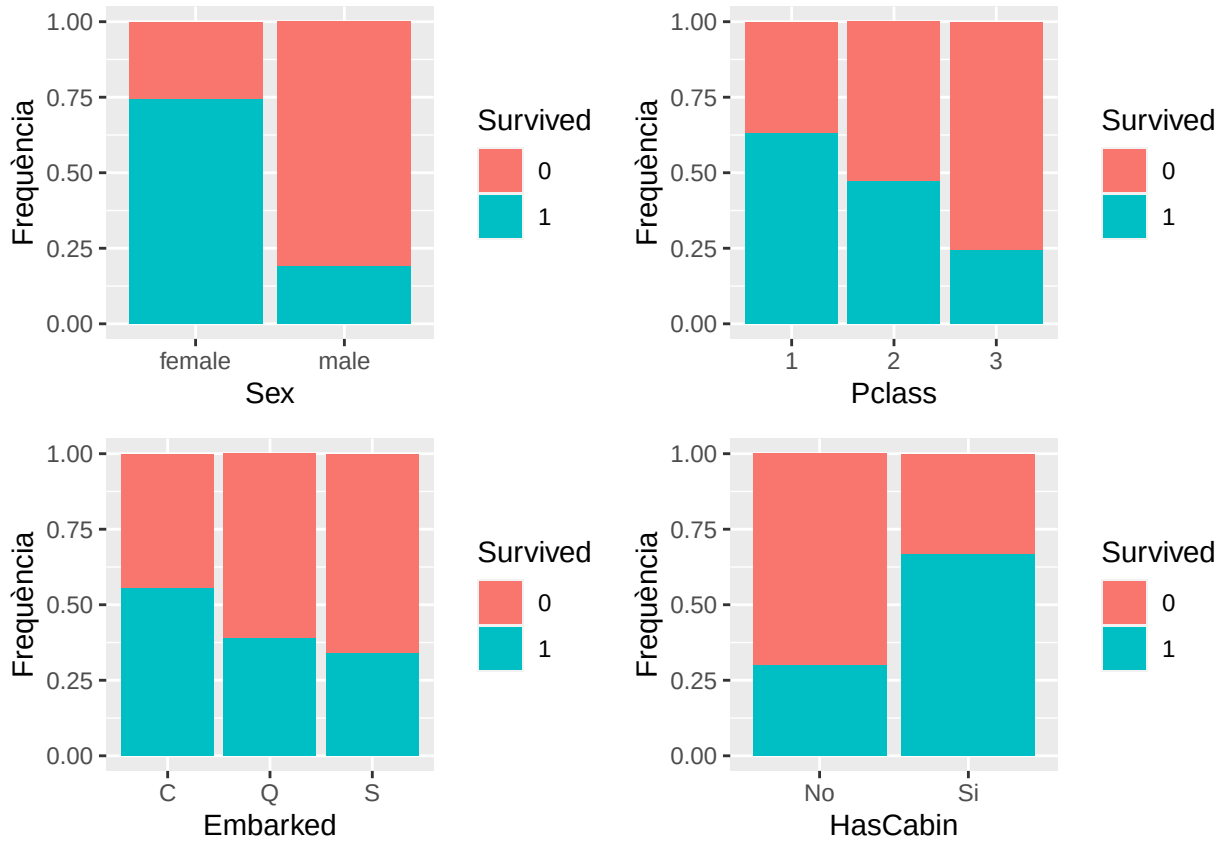
Un cop ja hem netejat les dades, podem començar amb l'anàlisi d'aquestes. Per a la anàlisis només utilitzem el dataset train, ja que és del que disposem tots els atributs necessaris per a poder valorar correctament els resultats.

2.4.1 Selecció dels grups de dades que es volen analitzar/comparar

L'objectiu és predir a partir dels atributs seleccionats si els passatgers van sobreviure o no a l'accident del titanic, anterior ment ja hem fet una selecció dels atributs que volem utilitzar. Per a la selecció dels grups de dades, seleccionem totes les dades del dataset train i les volem comparar en front de l'atribut *survived* per saber si ens aporten informació rellevant.

Comencem analitzant les diferents variables qualita

```
#careguem llibreries
library(ggplot2)
library(gridExtra)
# Visualitzem la relació entre les variables "sex" i "survival":
g1<-ggplot(data=train,aes(x=Sex,fill=Survived))+geom_bar(position="fill")+ylab("Frequència")
g2<-ggplot(data=train,aes(x=Pclass,fill=Survived))+geom_bar(position="fill")+ylab("Frequència")
g3<-ggplot(data=train,aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+ylab("Frequència")
g4<-ggplot(data=train,aes(x=HasCabin,fill=Survived))+geom_bar(position="fill")+ylab("Frequència")
grid.arrange(g1,g2,g3,g4,nrow=2)
```



De les diferents gràfiques podem veure com si que ens aporten informacio rellevant. Per exemple de la variable Sex, les dones tenen una probabilitat més alta de sobreviure que els homes, per a *Pclass* els de 1a classe tenen una probabilitat més alta que els de 2a que també tenen una probabilitat més alta que els de 3a, Els passatgers del por C tenen una probabilitat més alta de sobreviure que els dels altres 2 ports i els passatgers amb cabina tenen una probabilitat molt més alta de sobreviure que els que no en tenen.

2.4.2 Comprovació de la normalitat i homogeneitat de la variància.

Abans d'evaluar la relació que hi ha entre les variables quantitatives i la variable *Survived* hem dobtenir més informació d'aquestes per a saber quin és el millor metode per aplicar. Per això començem coprovant si questes variables segueixen una distribució normal o no. Per a fer-ho utilitzem el test de *Shapiro-Wilk*, que és considerat un dels mètodes més potents per contrastar la normalitat. Aquest mètode asumeix coma hipòtesi nul · la que la població està distribuida normalment, per tant si el p-valor és més petit que el nivell de significació (prendrem un valor de alpha = 0,05) llavors rebutjem la hipòtesi nul · la i per tant les dades no segueixen una distribució normal.

```
# fem el test de Sapiro-Wilk a les variables numèriques
alpha =0.05
col.names = colnames(train)
for (i in 1:ncol(train)) {
  #Comprovar si es numéric
  if (is.integer(train[,i]) | is.numeric(train[,i])) {
    #Fer el test
    f<-shapiro.test(train[,i])
    print(f)
    p_val =f$p.value
```

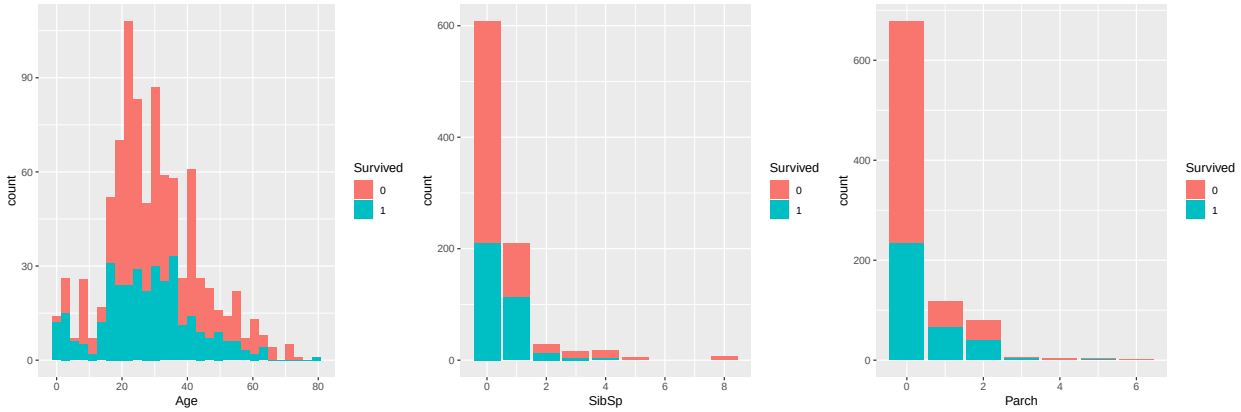
```
    cat(col.names[i])
    if (p_val < alpha) {
      cat(" no segueix distribució normal:\n")
    } else {
      cat(" segueix distribució normal:\n")
    }
  }
}
```

```
##
##  Shapiro-Wilk normality test
##
## data:  train[, i]
## W = 0.98005, p-value = 1.099e-09
##
## Age no segueix distribució normal:
##
##  Shapiro-Wilk normality test
##
## data:  train[, i]
## W = 0.51297, p-value < 2.2e-16
##
## SibSp no segueix distribució normal:
##
##  Shapiro-Wilk normality test
##
## data:  train[, i]
## W = 0.53281, p-value < 2.2e-16
##
## Parch no segueix distribució normal:
```

Tots els p-valors són pràcticament 0, per tant rebutgem les hipòtesis nul · les i assumim que no segueixen una distribució normal. Si les representem gràficament amb un histograma també veiem que la seva forma tampoc és la d'una distribució normal.

```
g1<-ggplot(data=train,aes(x=Age,fill=Survived))+geom_histogram()
g2<-ggplot(data=train,aes(x=SibSp,fill=Survived))+geom_bar()
g3<-ggplot(data=train,aes(x=Parch,fill=Survived))+geom_bar()
grid.arrange(g1,g2,g3,nrow=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



El seugent pas és comporvar l'homoscedesticitat en les dades, és a dir, de la igualtat de variàncies entre els grups que s'han de comparar. Com que les dades no segueixen una distribució nomral no podem aplicar el test de *Levene* i hem d'aplicar l'alternativa no paramètica que és el test de *Fligner-killeen*. Aquest mètode assumeix com a hipòtesi nul · la la igaltat de variàncies en els diferents grups de dades, de manera que p-valors inferiors al nivell de significació indicaran heteroscedesticitat.

```
# fem el test de Sapiro-Wilk a les variables numèriques
alpha =0.05
col.names = colnames(train)
for (i in 1:ncol(train)) {
  #Comprovar si es numéric
  if (is.integer(train[,i]) | is.numeric(train[,i])) {
    #Fer el test
    f<-fligner.test(x=list(train[,i],train$Survived))
    print(f)
    p_val =f$p.value
    cat(col.names[i])
    if (p_val < alpha) {
      cat(" no hi ha igualtat de variàncies:\n")
    } else {
      cat(" hi ha igualtat de variàncies:\n")
    }
  }
}
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(train[, i], train$Survived)
## Fligner-Killeen:med chi-squared = 1025.1, df = 1, p-value < 2.2e-16
##
## Age no hi ha igualtat de variàncies:
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(train[, i], train$Survived)
## Fligner-Killeen:med chi-squared = 1.7583, df = 1, p-value = 0.1848
##
## SibSp hi ha igualtat de variàncies:
##
```

```
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(train[, i], train$Survived)
## Fligner-Killeen:med chi-squared = 1.2757, df = 1, p-value = 0.2587
##
## Parch hi ha igualtat de variàncies:
```

Em obtingut resultats diferents, per a l'atribut *Age* hem obtingut un p-valor de pràcticament 0, per tant no hi ha igualtat de variàncies, en canvi, per als atributs *SibSp* i *Parch* si que hem obtingut igualtat de variàncies.

2.5. Aplicació de proves estadístiques per comparar els grups de dades.

2.5.1 Correlacions

El primer que fem és analitzar les correlacions entre la variable objectiu (*Survived*) i la resta de variables disponibles per determinar quines d'aquestes són les que exerceixen una major influencia. A l'apartat anterior hem vist que hi ha una relació, ara la quantificarem. Per a les variables numèriques, al no complir-se el criteri de normalitat i en el *Age* tampoc el d'homoscedasticitat, i al tenir una variable objectiu qualitativa, tindrem que utilitzar el test de *Kruskal-Wallis*, i per a les variables qualitatives utilitzarem el *Chi-Square test of independence*.

```
for (i in 1:(ncol(train))) {
  if(col.names[i]!="Survived"){
    print(col.names[i])
    if (is.integer(train[,i]) | is.numeric(train[,i])) {
      fun=kruskal.test(g=train[,i],x=train$Survived)
      print(fun)
    } else {
      tbl = table(train[,i],train$Survived)
      fun= chisq.test(tbl)
      print(fun)
      print(sqrt(fun$statistic / sum(tbl)))
    }
  }
}
```

```
## [1] "Pclass"
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 102.89, df = 2, p-value < 2.2e-16
##
## X-squared
## 0.3398174
## [1] "Sex"
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tbl
## X-squared = 260.72, df = 1, p-value < 2.2e-16
```

```
##
## X-squared
## 0.5409359
## [1] "Age"
##
##  Kruskal-Wallis rank sum test
##
## data:  train$Survived and train[, i]
## Kruskal-Wallis chi-squared = 185.78, df = 87, p-value = 3.911e-09
##
## [1] "SibSp"
##
##  Kruskal-Wallis rank sum test
##
## data:  train$Survived and train[, i]
## Kruskal-Wallis chi-squared = 37.23, df = 6, p-value = 1.588e-06
##
## [1] "Parch"
##
##  Kruskal-Wallis rank sum test
##
## data:  train$Survived and train[, i]
## Kruskal-Wallis chi-squared = 27.894, df = 6, p-value = 9.836e-05
##
## [1] "Embarked"
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 25.964, df = 2, p-value = 2.301e-06
##
## X-squared
## 0.1707068
## [1] "HasCabin"
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tbl
## X-squared = 87.941, df = 1, p-value < 2.2e-16
##
## X-squared
## 0.3141652
```

Observant els diferents resultat que hem obtingut, podem veure que en tots els casos el p-valor és inferior a 0.05, cosa que indica que podem rebutjar la hipòtesi nul · la de que les distribucions de grups de dades són les mateixes, i podem assumir que hi ha diferencies estadísticament significatives entre els grups de dades analitzades. És a dir hi ha una certa dependència entre les 2 variables. Per saber quina variable te una relació més forta que les altres ens fixem amb el valor *X-squared*, com més gran, més forta és la relació. així que les podem ordenar de més grana més petites i obtenim: *Sex* (260.72)> *Age* (185.78)> *Pclass* (102.89)> *HasCabin* (87.941)> *SibSp* (37.23)> *Parch* (27.894)> *Embarked* (25.964) Així dons la variable que té una relació més amb la supervivència del passatger és el sexe, seguit de l'edat i de la classe. Això te sentit amb la típica frase de les pel · lícules: "Les dones y els nens primer", juntament amb que els de primera classe tenien un poder i importància més elevada que els de 3a classe.

2.5.2 Comparació entre grups.

Ja sabem que hi ha una relació entre l'edat dels passatgers i la seva supervivència, però no sabem cap on es decanta aquesta relació. Per tant una de les preguntes que ens podríem fer és: L'edat dels supervivents és inferior a la dels no supervivents? Per resoldre aquesta pregunta farem un contrast d'hipòtesis sobre dos mostres. Hem de destacar que com que les dades no segueixen una distribució normal, tindríem que utilitzar un test no paramètric com el de *Mann-Whitney*, però ja que la nostra mostra és superior a 30 registres podem utilitzar l'aproximació de *t-student* per a fer el contrast. Així dons tenim com a hipòtesis nul · la que no hi ha diferencia entre la mitja d'edat entre els supervivents i els no supervivents i com a hipòtesi alternativa que la mitjana d'edat dels supervivent és menor.

H0 : u1 - u2 = 0 H1 : u1 - u2 < 0

```
  edatS = train[which(train$Survived=="1"),"Age"]
  edatNS = train[which( train$Survived=="0"),"Age"]
  t.test(edatS, edatNS,alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  edatS and edatNS
## t = -2.7448, df = 707.46, p-value = 0.003104
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -1.063947
## sample estimates:
## mean of x mean of y
##  27.66863  30.32878
```

Obtenim un p-valor de 0.0031, al ser inferior que 0.05 podem rebutjar la hipòtesi nul · la i acceptar l'alternativa de que l'edat dels supervivent és menor a la dels no supervivents. Això podria confirmar que més gent jove va sobreviure a l'accident del titànic. Una altre de les preguntes que ens podríem fer seria si la mitjan d'edat dels passatgers de sexe masculí que van sobreviure és més gran que al passatgers de sexe femení que van sobreviure? Fem un altre contrast d'hipòtesi aquest cop tenim: H0 : u1 - u2 = 0 H1 : u1 - u2 > 0

```
  edatSM = train[which(train$Sex=="male" & train$Survived=="1"),"Age"]
  edatSF = train[which(train$Sex=="female" & train$Survived=="1"),"Age"]
  t.test(edatSM, edatSF,alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  edatSM and edatSF
## t = -0.2201, df = 190.25, p-value = 0.587
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -3.232681       Inf
## sample estimates:
## mean of x mean of y
##  27.40982  27.78970
```

en aquest cas el p-valor és de 0.587 que és superior a 0.05, per tant no podem descartar la hipòtesi nul · la i sembla que no hi ha diferencia entre la mitjana d'edat dels passatger sobrevivents de sexe masculí i els de sexe femení.

2.5.3 Regressió lineal

Tot seguit intentarem crear un model de regressió lineal que utilitzi tant les variables qualitatives com quantitatives per poder fer les prediccions de si el passatger va sobreviure o no. Com que la variable a predir no és quantitativa tindrem que utilitzar la funció *glm* indicant que es binomial(), per a així transformar els possibles resultats a un resultat booleà. començarem creant diferents models per a veure com va afectant cada variable en el model i ens quedarem amb el que tingui un valor AIC (Akaike's Information Criteria) menor i de mica en mica anar sumant noves variables al millor model fins que ja no es pugui millorar. Així veurem si hi ha molta diferencia entre crear un model amb totes les variables o construir un model mica en mica per intentar obtenir el millor resultat. Comencem amb els models individuals per veure si les variables amb major correlació també produeixen models millors.

```
regS<-glm(Survived ~Sex,binomial(),train)
cat("Sex: ",regS$aic,"params:", regS$coefficients,"\n")
```

```
## Sex:  921.8039 params: 1.056589 -2.51371
```

```
regA<-glm(Survived ~Age,binomial(),train)
cat("Age: ",regA$aic,"params:", regA$coefficients,"\n")
```

```
## Age:  1182.98 params: -0.07332336 -0.01380151
```

```
regP<-glm(Survived ~Pclass,binomial(),train)
cat("Pclass: ",regP$aic,"params:", regP$coefficients,"\n")
```

```
## Pclass:  1089.108 params: 0.5306283 -0.6394311 -1.670399
```

```
regC<-glm(Survived ~HasCabin,binomial(),train)
cat("HasCabin: ",regC$aic,"params:", regC$coefficients,"\n")
```

```
## HasCabin:  1102.856 params: -0.8479911 1.541138
```

```
regSi<-glm(Survived ~SibSp,binomial(),train)
cat("SibSp: ",regSi$aic,"params:", regSi$coefficients,"\n")
```

```
## SibSp:  1189.515 params: -0.4381535 -0.06863757
```

```
regPa<-glm(Survived ~Parch,binomial(),train)
cat("Parch: ",regPa$aic,"params:", regPa$coefficients,"\n")
```

```
## Parch:  1184.842 params: -0.5530505 0.2033171
```

```
regE<-glm(Survived ~Embarked,binomial(),train)
cat("Embarked: ",regE$aic,"params:", regE$coefficients,"\n")
```

```
## Embarked:  1167.291 params: 0.2151114 -0.6640616 -0.8828237
```

Efectivament, sembla que la variable *Sex* és la que obté un millor resultat, tot i que després la variable *PClass* i *HasCabin* obtenen un millor resultat que *Age*. Provem afegint una segona variable al model amb la variable *Sex*:

```
reg<-glm(Survived ~Sex+Age,binomial(),train)
cat("Sex+Age: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+Age:  923.3353 params: 1.165995 -2.499379 -0.004063182
```

```
reg<-glm(Survived ~Pclass+Age,binomial(),train)
cat("Sex+Pclass: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+Pclass:  1025.447 params: 2.524147 -1.157648 -2.5101 -0.04972677
```

```
reg<-glm(Survived ~Sex+HasCabin,binomial(),train)
cat("Sex+HasCabin: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin:  851.8062 params: 0.6881938 -2.580295 1.664113
```

```
reg<-glm(Survived ~Sex+SibSp,binomial(),train)
cat("Sex+SibSp: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+SibSp:  910.6925 params: 1.28373 -2.639602 -0.2978931
```

```
reg<-glm(Survived ~Sex+Parch,binomial(),train)
cat("Sex+Parch: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+Parch:  920.428 params: 1.185953 -2.603052 -0.1865116
```

```
reg<-glm(Survived ~Sex+Embarked,binomial(),train)
cat("Sex+Embarked: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+Embarked:  906.6944 params: 1.784631 -2.533934 -1.061566 -0.8725014
```

De les combinacions provades sembla que el millor model és el creat per les variables (*Sex+HasCabin*) amb un valor AIC de 851.8062, que millora el model de la variable *Sex* sola, seguit de la combinació amb *Embarked, SibSp, Parch, Age* i *Pclass*. Per contra del que veiem anteriorment, sembla que el model que te 2 de les variables amb més correlació no és el millor, això segurament és degut a que la informació que ens aporten les 2 variables es redundant, en canvi el model amb la variable *Sex+HasCabin* conté menys informació redundant que ajuda a classificar millor les dades.

Seguim afegint una tercera variable.

```
reg<-glm(Survived ~Sex+HasCabin+Age,binomial(),train)
cat("Sex+HasCabin+Age: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Age:  841.2025 params: 1.254952 -2.508037 1.924527 -0.02322473
```

```
reg<-glm(Survived ~Sex+HasCabin+Pclass,binomial(),train)
cat("Sex+HasCabin+Pclass: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass:  828.3345 params: 1.543346 -2.625888 0.9255961 -0.1678766 -1.187733
```

```
reg<-glm(Survived ~Sex+HasCabin+SibSp,binomial(),train)
cat("Sex+HasCabin+SibSp: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+SibSp:  843.0533 params: 0.9097682 -2.700602 1.65014 -0.2835718
```

```
reg<-glm(Survived ~Sex+HasCabin+Parch,binomial(),train)
cat("Sex+HasCabin+Parch: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Parch:  850.0418 params: 0.8277905 -2.677505 1.67932 -0.1972368
```

```
reg<-glm(Survived ~Sex+HasCabin+Embarked,binomial(),train)
cat("Sex+HasCabin+Embarked: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Embarked:  847.7346 params: 1.210056 -2.579435 1.575944 -0.5532737 -0.6299627
```

Sembla que seguim millorant el model ja que hem obtingut un valor de AIC de 828.3345 amb la combinació de (*Sex+HasCabin+PClass*), seguit de la combinació amb *Age, SibSp, Embarked* i *Parch*.

Seguim afegint una variable més al millor model.

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Age,binomial(),train)
cat("Sex+HasCabin+Pclass+Age: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Age:  799.6724 params: 3.148319 -2.528427 0.8392392 -0.660776 -1.934995 -0.039633
```

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+SibSp,binomial(),train)
cat("Sex+HasCabin+Pclass+SibSp: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+SibSp:  822.3425 params: 1.701413 -2.728521 0.9458132 -0.1602158 -1.134237 -0.25
```

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Parch,binomial(),train)
cat("Sex+HasCabin+Pclass+Parch: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Parch:  827.7645 params: 1.620264 -2.705915 0.9744944 -0.1328332 -1.141738 -0.16
```

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Embarked,binomial(),train)
cat("Sex+HasCabin+Pclass+Embarked: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Embarked:  823.5593 params: 1.871838 -2.595807 0.9833668 0.04764457 -1.078515 -0
```

Sembla que ara si que la variable *Age* és la que ens ajuda a millorar més el model actual amb un AIC de 799.672. Seguim l'addició:

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Age+SibSp,binomial(),train)
cat("Sex+HasCabin+Pclass+Age+SibSp: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Age+SibSp:  780.7809 params: 3.862439 -2.676669 0.8471952 -0.7945216 -2.040398 -
```

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Age+Parch,binomial(),train)
cat("Sex+HasCabin+Pclass+Age+Parch: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Age+Parch:  796.3463 params: 3.358215 -2.627502 0.9010764 -0.6358663 -1.906108 -
```

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Age+Embarked,binomial(),train)
cat("Sex+HasCabin+Pclass+Age+Embarked: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Age+Embarked:  796.828 params: 3.358985 -2.487813 0.8907573 -0.4534256 -1.825515
```

Aquest cop la variable *SibSp* és la que ens ajuda a millorar una mica més amb un AIC de 780.7809.

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Age+SibSp+Parch,binomial(),train)
cat("Sex+HasCabin+Pclass+Age+SibSp+Parch: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Age+SibSp+Parch:  782.1688 params: 3.889004 -2.706652 0.8680133 -0.7761447 -2.02
```

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Age+SibSp+Embarked,binomial(),train)
cat("Sex+HasCabin+Pclass+Age+SibSp+Embarked: ",reg$aic,"params:", reg$coefficients,"\n")
```

```
## Sex+HasCabin+Pclass+Age+SibSp+Embarked:  780.8983 params: 3.987988 -2.632446 0.8854378 -0.6199995 -1
```

La resta de variables tan *Parch* com *Embarked* no ens ajuden a millorar el model, ja que la puntuació AIC no millora. Així dons podem descartar aquestes 2 variables ja que no ens són d'utilitat per el model de regressió lineal que hem obtingut.

```
reg<-glm(Survived ~Sex+HasCabin+Pclass+Age+SibSp,binomial(),train)
summary(reg)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + HasCabin + Pclass + Age + SibSp,
##     family = binomial(), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8961  -0.5761  -0.3749   0.6009   2.5169
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.862439   0.506501   7.626 2.43e-14 ***
## Sexmale     -2.676669   0.197233 -13.571  < 2e-16 ***
## HasCabinSi   0.847195   0.332818   2.546   0.0109 *
## Pclass2     -0.794522   0.367121  -2.164   0.0304 *
## Pclass3     -2.040398   0.372976  -5.471 4.49e-08 ***
## Age         -0.050439   0.008088  -6.236 4.48e-10 ***
## SibSp       -0.430274   0.105696  -4.071 4.68e-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:  766.78  on 884  degrees of freedom
## AIC: 780.78
##
## Number of Fisher Scoring iterations: 5
```

```
taula <- table(train$Survived,predict(object=reg, newdata =train, type="response")> 0.5)
taula
```

```
##
##     FALSE TRUE
##   0   474   75
##   1    89  253
```

```
precisio <- sum(diag(taula)) / sum(taula)
precisio
```

```
## [1] 0.8159371
```

```
cat("error:",(1-precisio)*100,"%")
```

```
## error: 18.40629 %
```

Si directament haguéssim fet un model amb totes les variables que teníem ja que semblava que ens aportaven informació haguéssim tingut el model:

```
reg<-glm(Survived ~Sex+Age+Pclass+HasCabin+SibSp+Parch+Embarked,binomial(),train)
summary(reg)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Age + Pclass + HasCabin + SibSp +
##     Parch + Embarked, family = binomial(), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7780  -0.5668  -0.3760   0.6103   2.5435
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.020203   0.518632   7.752 9.08e-15 ***
## Sexmale     -2.663962   0.204419 -13.032  < 2e-16 ***
## Age         -0.048966   0.008115  -6.034 1.60e-09 ***
## Pclass2     -0.601543   0.380322  -1.582 0.113725
## Pclass3     -1.911963   0.384408  -4.974 6.57e-07 ***
## HasCabinSi   0.906080   0.336006   2.697 0.007005 **
## SibSp       -0.378251   0.110568  -3.421 0.000624 ***
## Parch       -0.088689   0.120378  -0.737 0.461270
## EmbarkedQ   -0.074905   0.403819  -0.185 0.852843
```

```
## EmbarkedS   -0.435306   0.242383  -1.796 0.072504 .
## ---
## 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:  762.34  on 881  degrees of freedom
## AIC: 782.34
##
## Number of Fisher Scoring iterations: 5
```

```
taula <- table(train$Survived,predict(object=reg, newdata =train, type="response")> 0.5)
taula
```

```
##
##     FALSE TRUE
##   0   480   69
##   1    90  252
```

```
precisio <- sum(diag(taula)) / sum(taula)
precisio
```

```
## [1] 0.8215488
```

```
cat("error:",(1-precisio)*100,"%")
```

```
## error: 17.84512 %
```

Els dos models són molt similars, tot i que el model amb menys variables te un AIC una mica millor de 780.78 en front del 782.34 del model amb totes les variable. Però en el cas concret de les dades que li hem passat la precisió del model amb totes les dades ha sigut una mica superior amb un error del 17.85% en front de 18.4% del model anterior.

2.5.3 Random Forest Classifie

Els models de regressió tenen un valor AIC és molt elevat, i un error al voltant del 18%. Per tant el model de regressió lineal, tot i ser el millor que hem pogut obtenir, no s'ajusta gaire bé. Així que podem provar altres models a veure si s'ajusten millor, per exemple podem provar un model RandomForestClassifier:

```
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:gridExtra':
##
##     combine

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

```
set.seed(51)
rf<-randomForest(Survived~.,data = train,method = 'rf',trControl = trainControl(method = 'cv',number = 5
rf
```

```
##
## Call:
##  randomForest(formula = Survived ~ ., data = train, method = "rf",      trControl = trainControl(meth
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.5%
## Confusion matrix:
##     0   1 class.error
## 0 512  37  0.06739526
## 1 110 232  0.32163743
```

Provem també de crear el model amb les millors variables que em trobat amb el model de regressió lineal, per comprovar si hi ha molt diferencia:

```
set.seed(51)
rf<-randomForest(Survived~Sex+Age+Pclass+HasCabin+SibSp,data = train,method = 'rf',trControl = trainCon
rf
```

```
##
## Call:
##  randomForest(formula = Survived ~ Sex + Age + Pclass + HasCabin +      SibSp, data = train, method =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.05%
## Confusion matrix:
##     0   1 class.error
## 0 517  32   0.0582878
## 1 111 231   0.3245614
```

Els 2 models obtinguts són millors que els de regressió lineal, i en aquest cas, el model amb menys variables també té una millor precisió que el model amb totes les variables, Tot i que continua sent un valor elevat del 16.05% d'error.

2.6. Conclusions

Com hem vist, sempre s'ha de fer un pretractament a tots els datasets que s'obtenen per a fer una correcció d'errors, normalització i estandardització, que ajuden a facilitar la feina posteriorment. També permet

extreure informació inicial del dataset, com per exemple quines variables no aporten informació per a la resolució del problema i així eliminar-les. També hem vist que es poden utilitzar diferents mètodes per a corregit elements buits, com pot ser imputació de valors o eliminació de les dades, i com els valors extrems també poden aportar informació i no sempre s'han d'eliminar.

Posteriorment al tractament de dades, hem fet un anàlisi de correlacions per a veure de les variables restants quines ens aportaven més informació a l'hora de resoldre el problema. També ens hem plantejat diferents preguntes que és poden resoldre amb les dades disponibles per així extreure més coneixement del dataset que ens pugui ser útil per a la resolució del problema inicial.

Finalment, hem intentat crear un model de regressió lineal i un model random forest que ens ajudessin a donar resposta al problema inicial de descobrir si els passatgers havien sobreviscut o no al accident del titànic. Amb les dades disponibles no hem pogut crear de manera simple un model que ens dones una solució molt acurada, el millor que hem obtingut ha sigut un model amb un error de 16.05%. Segurament amb models més complexos i quer requereixin un nivell de computació més alt, podríem intentar aconseguir un model més acurat.