PRÀCTICA 2: Neteja i validació de les dades

Marc Guerrero Molero

3 de January, 2020

Índex

1. Detalls de l'activitat **1**
1.1. Descripció . . . 1
1.2. Objectius . . . 1
1.3. Competències . . . 2

2. Resolució **2**
2.1. Descripció del dataset . . . 2
2.2. Integració i selecció de les dades d'interès a analitzar. . . . 2

3. Neteja de dades **4**
3.1. Cerca de zeros i elements buits . . . 4
3.2. Valors extrems . . . 5
3.3. Exportació de les dades . . . 9

4. Anàlisi de dades **9**
4.1. Selecció dels grups de dades . . . 9
4.2. Comprovació de la normalitat i homogeneïtat de la variància. . . . 9

5. Aplicació de proves estadístiques **12**
5.1. Matriu de correlació . . . 12
5.2. Contrast d'hipòtesis . . . 17
5.3. Regressió logistica . . . 19
5.4. Random forest . . . 25

6. Presentació de resultats **27**

7. Conclusions **29**

8. Referències **30**

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 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

Les competències del màster de Data Science que es desenvolupen en aquesta pràctica són:

- 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

2.1. Descripció del dataset

El conjunt de dades que es farà servir en aquesta pràctica es tracta del Titànic obtingut de la web *Kaggle*. Aquest *dataset* tracta sobre característiques d'individus que anaven al Titànic on l'objectiu és predir si aquests han sigut capaços de sobreviure a l'accident.

En els camps del joc de dades trobem els següents:

- **Survival**: 1 en cas que el passatger ha sobreviscut, 0 en cas contrari.
- **Pclass**: Classe socioeconòmica del passatger. Hi ha tres classes 1: 1st (alta), 2: 2nd (mtijana), 3: 3rd (Baixa).
- **Sex**: Sexe del passatger.
- **Age**: Edat en anys del passatger.
- **Sibsp**: Nombre de germans, marits i mullers a bord.
- **Parch**: Nombre de fills i pares a bord.
- **Ticket**: Número del tiquet.
- **Fare**: Tarifa del passatger.
- **Cabin**: Número de cabina.
- **Embarked**: Port de l'embarcació Hi ha tres tipus: C = Cherbourg, Q = Queenstown, S = Southampton.

Amb l'anàlisi d'aquest joc de dades es pretrén veure quins factors tenen major importància en la supervivència d'una persona en l'accident del Titànic i fer prediccions utilitzant models sobre si un individu sobreviuria en un accident així.

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

En aquest apartat es llegiran les dades i se seleccionaran aquells atributs que siguin rellevant per l'anàlisi.

```r
# Lectura de dades
df_titanic <- read.csv("../data/titanic.csv", header = TRUE)
head(df_titanic)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex Age 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
## 6                                   Moran, Mr. James   male  NA     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
## 6           330877  8.4583              Q
```

Els atributs que no es faran servir són:

- PassengerId: No dóna cap informació útil sobre el passatger.
- Name: No dóna cap informació al futur anàlisi.
- Cabin: Com que no se sap la disposició de les cabines dins del vaixell, aquest paràmetre no aporta informació útil per l'anàlisi.
- Ticket: El número del tiquet no aporta informació en l'anàlisi.

```r
# Esborrem els paràmetres que no es fan servir
del_col <- c("PassengerId", "Name", "Cabin", "Ticket")
df_titanic <- df_titanic[ , !(names(df_titanic) %in% del_col)]
```

Un cop feta la selecció de les dades, s'ha de mirar si els tipus de dades assignats automàticament per l'R són correctes.

```r
# Mirem el tipus de les dades
str(df_titanic)
```

```
## 'data.frame':    891 obs. of  8 variables:
##  $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ 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 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Com es pot veure, el format de les dades és el correcte.

3. Neteja de dades

3.1. Cerca de zeros i elements buits

En aquest cas, els valors buits estan marcats amb NA, ja que el valor 0 no pot ser utilitzat com a sentinella perquè aquest sí que té un significat en els atributs. Per això, buscarem els registres que contenen NA com a valor.

```
# Mirem valors buits
sapply(df_titanic, function(x) sum(is.na(x)))
```

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

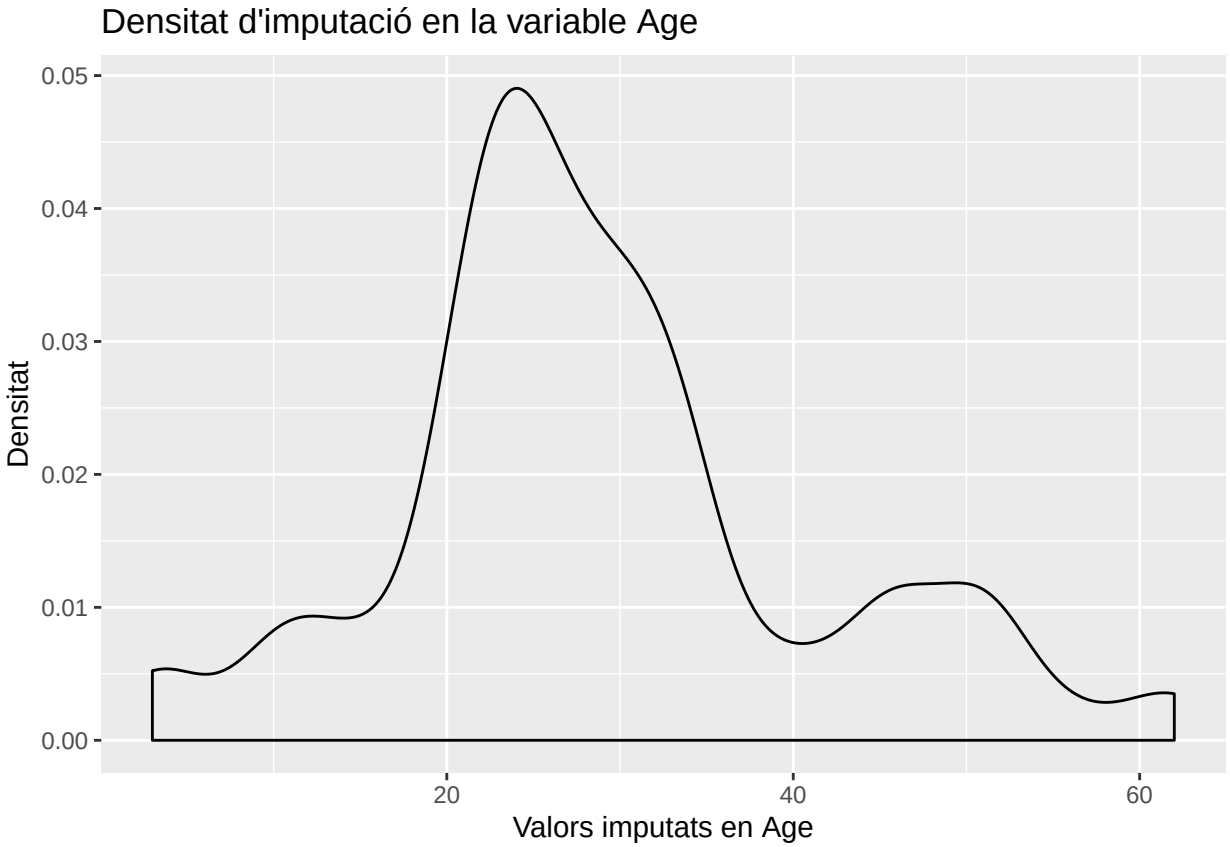
Com es pot observar, en la variable *Age* hi ha 177 registres que contenen el camp buit. Això representa un 20% dels registres, per tant, no es pot esborrar, ja que és un alt percentatge del conjunt de dades. Pel tractament dels valors NA es farà servir l'algoritme *k-nearest neighbors* per la imputació del valor.

```
# Utilitzem la funció kNN del paquet vIM
knn_df <- kNN(df_titanic)
df_titanic$Age <- knn_df$Age
# comprovem que aquest han estat tractats
sapply(df_titanic, function(x) sum(is.na(x)))
```

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

Un cop vist que no hi ha valors buits en la variable *Age*, es pot observar quins són els valors imputats en aquest atribut per l'algoritme.

```
# Visualitzem els valors imputats per l'algoritme
ggplot(mapping= aes(x=knn_df$Age[knn_df$Age_imp == TRUE]))+ geom_density() +
labs(x = "Valors imputats en Age", title="Densitat d'imputació en la variable Age",
     y= "Densitat")
```

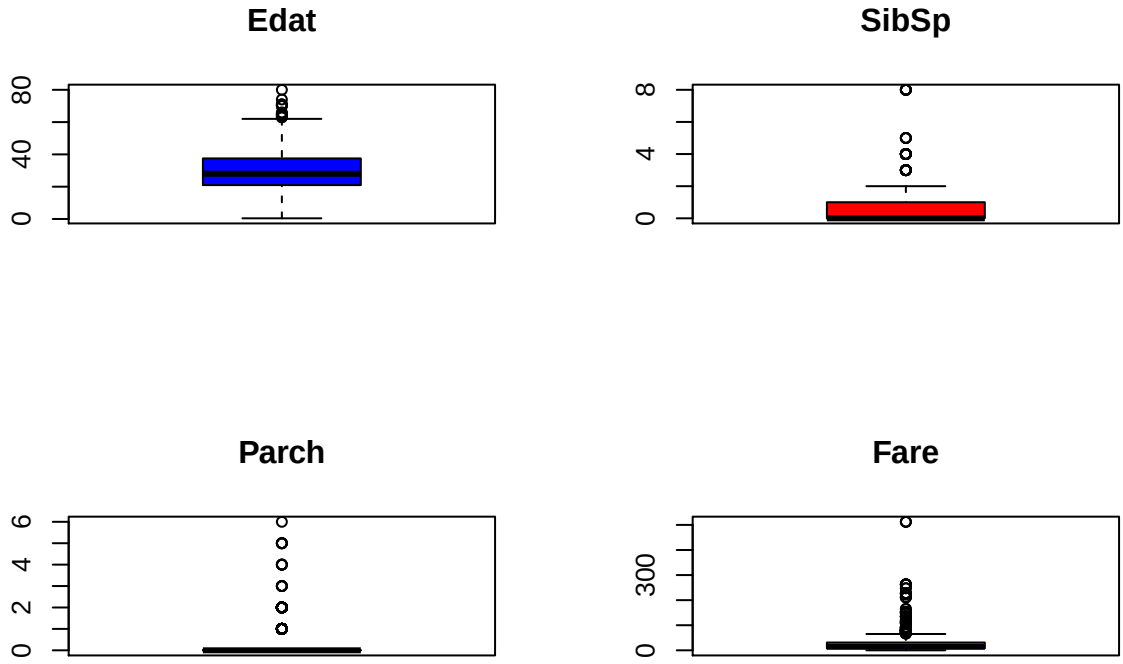


Com s'observa en la gràfica, els valors imputats amb més freqüència estan entre 20 i 30.

3.2. Valors extrems

En aquest apartat es buscaran els valors extrems dins del *dataset*, és a dir, els valors que semblen que no pertanyin a aquest joc de dades. Normalment, els considerats com *outliers* solen distar el seu valor en més de tres desviacions típiques respecte a la mitjana, per tant, s'utilitzaran diagrames de caixes per a detectar-los. En aquests només es representaran aquelles variables que siguin numèriques, és a dir, les variables *Age*, *SibSp*, *Parch* i *Fare*.

```
par(mfrow = c(2,2))
boxplot(df_titanic$Age,main="Edat", col="blue")
boxplot(df_titanic$SibSp,main="SibSp", col="red")
boxplot(df_titanic$Parch,main="Parch", col="green")
boxplot(df_titanic$Fare,main="Fare", col="orange")
```



Seguidament, es mostren els valor que han estat considerats com extrems.

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

```
##  [1] 66.0 65.0 71.0 70.5 63.0 65.0 64.0 65.0 63.0 71.0 64.0 80.0 70.0 70.0 74.0
```

```
boxplot.stats(df_titanic$SibSp)$out
```

```
##  [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3 5 4 3
## [39] 4 8 4 3 4 8 4 8
```

```
boxplot.stats(df_titanic$Parch)$out
```

```
##   [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1 2 1
##  [38] 1 2 1 4 1 1 1 1 2 2 1 2 1 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 1 1 1 1 1 2 1 2
##  [75] 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2 2 3 4 1 2 1
## [112] 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1 2 1 1 2 5 2 1 1
## [149] 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1 1 2 1 2 3 1 2 1 2 2
## [186] 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1 1 5 2
```

```
boxplot.stats(df_titanic$Fare)$out
```

```
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000
##   [9] 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000  66.6000  69.5500
##  [17]  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000  83.4750  90.0000
##  [25]  79.2000  86.5000 512.3292  79.6500 153.4625 135.6333  77.9583  78.8500
##  [33]  91.0792 151.5500 247.5208 151.5500 110.8833 108.9000  83.1583 262.3750
##  [41] 164.8667 134.5000  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000
##  [49] 263.0000  75.2500  69.3000 135.6333  82.1708 211.5000 227.5250  73.5000
```

```
##  [57] 120.0000 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042  91.0792
##  [65]  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000 221.7792 106.4250
##  [73]  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833  79.6500  79.2000
##  [81]  78.2667 153.4625  77.9583  69.3000  76.7292  73.5000 113.2750 133.6500
##  [89]  73.5000 512.3292  76.7292 211.3375 110.8833 227.5250 151.5500 227.5250
##  [97] 211.3375 512.3292  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583
## [105] 211.3375  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500
## [113]  89.1042 164.8667  69.5500  83.1583
```

Un cop vist aquests valors, es pot observar que els *outliers* en la variable edat entren dins del que es podria considerar rangs normals, ja que tots estan per sota dels vuitanta anys. Per tant, en aquest cas no s'aplica cap mesura.

En el cas de l'atribut SibSp, abans de prendre cap decisió, s'ha de tenir en compte els valors dels altres registres. Per a poder examinar-los, es miren els registres que tinguin un valor igual o superior a 5, ja que marcarien una família bastant gran i, per tant, podria significar un cas estrany.

```
df_titanic[df_titanic$SibSp >= 5,]
```

```
##     Survived Pclass    Sex Age SibSp Parch  Fare Embarked
## 60         0      3   male  11     5     2 46.90        S
## 72         0      3 female  16     5     2 46.90        S
## 160        0      3   male  11     8     2 69.55        S
## 181        0      3 female  11     8     2 69.55        S
## 202        0      3   male  11     8     2 69.55        S
## 325        0      3   male  11     8     2 69.55        S
## 387        0      3   male   1     5     2 46.90        S
## 481        0      3   male   9     5     2 46.90        S
## 684        0      3   male  14     5     2 46.90        S
## 793        0      3 female  11     8     2 69.55        S
## 847        0      3   male  11     8     2 69.55        S
## 864        0      3 female  11     8     2 69.55        S
```

Unes de les característiques que s'han de complir és que hi haguin el mateix més un nombre de registres amb el mateix valor, ja que si es té 8 germans en el mateix vaixell, hi ha d'haver 9 persones que tinguin 8 germans o germanastres. Si observem els valors anteriorment mostrats, es pot veure que no es compleix degut a que suposadament faltaría un membre de la família. Tot i així, es considera que més que una dada anòmala hi ha algun tipus de pèrdua d'informació en el *dataset*. També cal tenir en compte l'època de les dades, és a dir, avui en dia sería molt més anòmal trobar una família de vuit germans que anteriorment i, per tant, aquests casos s'han de considerar com factibles.

En conclusió, després de les observacions fetes, es pot concloure que aquests valors no són *outliers* i entren dins del rang de possibles ocurrències.

Seguidament, s'exploraran els valors extrems de la variable Parch amb un valor superior a 2. S'ha de tenir en compte que els valors extrems d'aquesta variable han de ser tractats amb certa precaució, ja que la majoria de registres contenen el valor 0 i, per tant, valors com 2, que poden marcar els dos pares, són considerats com *outliers*. Per tant, per a explorar els valors es tindran en compte els que siguin majors a 2.

```
df_titanic[df_titanic$Parch > 2,]
```

```
##     Survived Pclass    Sex Age SibSp Parch     Fare Embarked
## 14         0      3   male  39     1     5  31.2750        S
## 26         1      3 female  38     1     5  31.3875        S
## 87         0      3   male  16     1     3  34.3750        S
## 168        0      3 female  45     1     4  27.9000        S
## 361        0      3   male  40     1     4  27.9000        S
## 438        1      2 female  24     2     3  18.7500        S
```

```
## 439        0      1   male  64     1     4 263.0000        S
## 568        0      3 female  29     0     4  21.0750        S
## 611        0      3 female  39     1     5  31.2750        S
## 639        0      3 female  41     0     5  39.6875        S
## 679        0      3 female  43     1     6  46.9000        S
## 737        0      3 female  48     1     3  34.3750        S
## 775        1      2 female  54     1     3  23.0000        S
## 859        1      3 female  24     0     3  19.2583        C
## 886        0      3 female  39     0     5  29.1250        Q
```

Observant les dades, es pot veure que els valors iguals o més petits que 6 tenen una explicació bastant lògica considerant que en el vaixell poden viatjar els dos pares i entre dos i quatre fills. A més a més, si es compara amb l'edat, aquesta està dins de rangs on aquest tipus de família pot encaixar. Ara bé, hi ha un cas on el nombre de fills i pares és 3 i l'edat és de 16. Això implicaria viatjar amb els dos pares i un fill i, per tant, seria un cas molt anòmal tenir aquesta edat amb aquesta configuració de família. En conseqüència, s'esborrarà aquest registre.

```
df_titanic <- df_titanic[!(df_titanic$Parch == 3 & df_titanic$Age == 16),]
```

Per últim, en el preu del tiquet del vaixell es procedirà de la mateixa manera que en el cas anterior. Primerament, s'observen les dades.

```
df_titanic[df_titanic$Fare > 200,]
```

```
##     Survived Pclass    Sex Age SibSp Parch     Fare Embarked
## 28         0      1   male  19     3     2 263.0000        S
## 89         1      1 female  23     3     2 263.0000        S
## 119        0      1   male  24     0     1 247.5208        C
## 259        1      1 female  35     0     0 512.3292        C
## 300        1      1 female  50     0     1 247.5208        C
## 312        1      1 female  18     2     2 262.3750        C
## 342        1      1 female  24     3     2 263.0000        S
## 378        0      1   male  27     0     2 211.5000        C
## 381        1      1 female  42     0     0 227.5250        C
## 439        0      1   male  64     1     4 263.0000        S
## 528        0      1   male  24     0     0 221.7792        S
## 558        0      1   male  24     0     0 227.5250        C
## 680        1      1   male  36     0     1 512.3292        C
## 690        1      1 female  15     0     1 211.3375        S
## 701        1      1 female  18     1     0 227.5250        C
## 717        1      1 female  38     0     0 227.5250        C
## 731        1      1 female  29     0     0 211.3375        S
## 738        1      1   male  35     0     0 512.3292        C
## 743        1      1 female  21     2     2 262.3750        C
## 780        1      1 female  43     0     1 211.3375        S
```

Com es pot veure, hi ha molts valors entre 200 i 300. Això pot significar una compra d'última hora coincidint que tots pertanyen a primera classe socioeconòmica. Hi ha tres casos estranys on el valor és aproximadament el doble que el següent valor més alt. Aquest serà considerat anòmal i s'utilitzarà l'algoritme KNN per fer una imputació més justa d'aquest preu.

```
df_titanic[df_titanic$Fare > 500,"Fare"] = NA
knn_df <- kNN(df_titanic)
df_titanic$Fare <- knn_df$Fare
#visualitzem el nou valor imputat
knn_df[knn_df$Fare_imp == TRUE,"Fare"]
```

```
## [1] 26.55 79.65 26.55
```

Un cop imputat aquest valor, es dóna per finalitzat el tractament dels valors extrems.

3.3. Exportació de les dades

En aquest apartat, s'exporten les dades processades en els apartats anteriors.

```
# Exportació de les dades
write.csv(df_titanic, "../data/titanic_clean.csv")
```

4. Anàlisi de dades

4.1. Selecció dels grups de dades

En aquest apartat, se separaran diversos grups per a poder ser comparats entre ells. Aquests seran els següents:

- Homes vs Dones
- Nens (menors de 18), adults (entre 18 i 50) i gent gran (majors de 50)
- Classe socioeconòmica
- Tenen família vs no tenen família a bord
- Port d'embarcació

Els grups es dividiran en el moment de l'anàlisi, però si cal recodificar algunes variables i crear unes de noves per a complir amb els criteris anteriors.

```
#Nens, adults i gent gran
df_titanic$Age_group <- df_titanic$Age
df_titanic$Age_group[df_titanic$Age<18] = "children"
df_titanic$Age_group[df_titanic$Age>=18 & df_titanic$Age<50] = "adult"
df_titanic$Age_group[df_titanic$Age>=50] = "old"
df_titanic$Age_group <- as.factor(df_titanic$Age_group)

# Classe socioeconòmica
df_titanic$Pclass[df_titanic$Pclass == 1] = "upper"
df_titanic$Pclass[df_titanic$Pclass == 2] = "middle"
df_titanic$Pclass[df_titanic$Pclass == 3] = "lower"
df_titanic$Pclass <- as.factor(df_titanic$Pclass)

# Tenen família vs no tenen família a bord (nou atribut)
df_titanic$family <- df_titanic$Age
df_titanic$family[df_titanic$SibSp== 0 & df_titanic$Parch == 0] = "yes"
df_titanic$family[df_titanic$SibSp!= 0 | df_titanic$Parch != 0] = "no"
df_titanic$family <- as.factor(df_titanic$family)
```

Cal esmentar que la comparació final dels grups proposats dependrà dels resultats que s'obtinguin en els apartats següents.

Un cop separades les dades seguint els criteris establerts, s'ha de comprovar que aquestes són normals i examinar l'homogeneïtat de la variància.

4.2. Comprovació de la normalitat i homogeneïtat de la variància.

Per a la comprovació que cada variable quantitativa del *dataset* prové d'una població distribuïda normalment, s'utilitzarà el test de Shapiro-Wilk, ja que aquest és considerat un dels mètodes més potents per contrastar la

normalitat. Per a fer-ho, s'utilitzarà la funció *shapiro.test()* en cada variable quantitativa marcant el nivell de significació en 0.05.

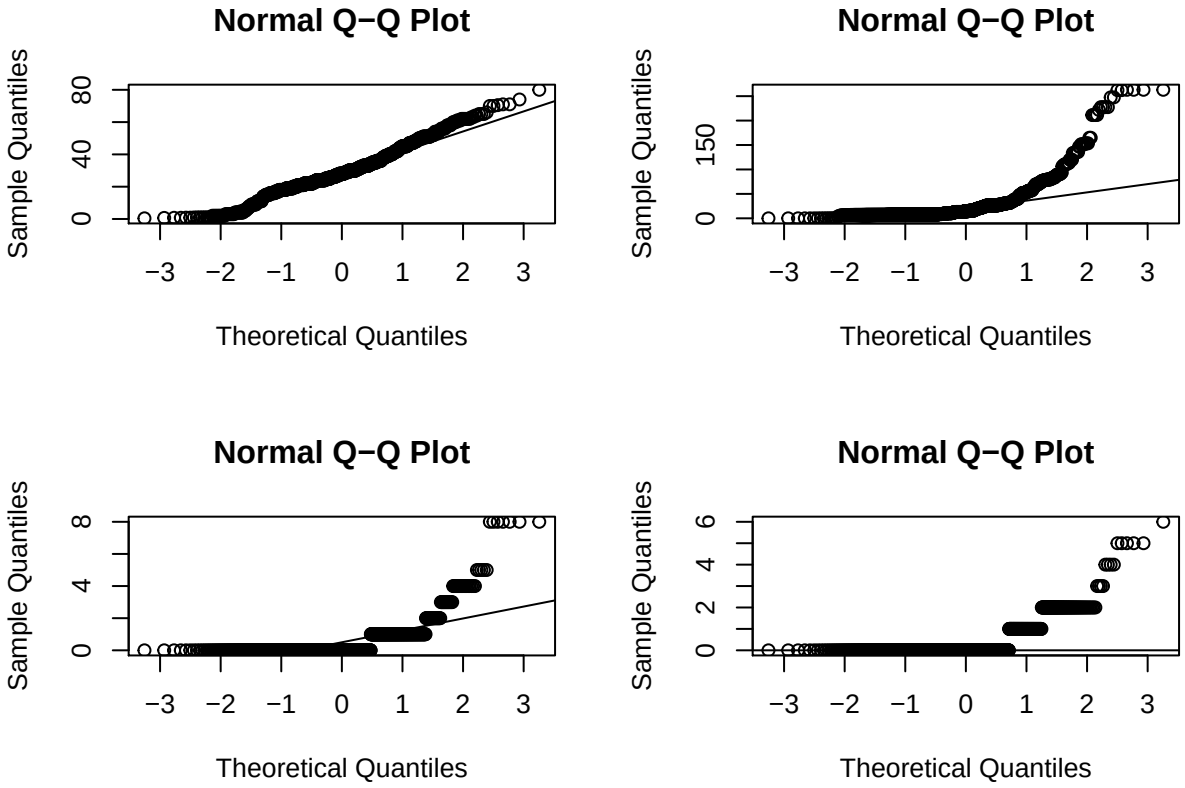
```
col <- c("Age", "Fare", "SibSp", "Parch")
result <- data.frame(
  "Age" = TRUE,
  "Fare" = TRUE,
  "SibSp" = TRUE,
  "Parch" = TRUE
)
for (i in col){
  if(shapiro.test(df_titanic[,i])$p.value < 0.05){
    result[1,i] = FALSE
  }
}
result
```

```
##     Age  Fare SibSp Parch
## 1 FALSE FALSE FALSE FALSE
```

Com marca la sortida del codi anterior, on es comprova si el resultat del test accepta que les dades segueixen una distribució normal, es pot veure que cap dels atributs anteriors ha donat positiu. Per tant, s'ha de considerar que no hi ha normalitat en les dades.

Per a veure-ho amb més claredat, es pot utilitzar eina visual com el gràfic QQ plot.

```
par(mfrow = c(2,2))
qqnorm(df_titanic$Age)
qqline(df_titanic$Age)
qqnorm(df_titanic$Fare)
qqline(df_titanic$Fare)
qqnorm(df_titanic$SibSp)
qqline(df_titanic$SibSp)
qqnorm(df_titanic$Parch)
qqline(df_titanic$Parch)
```



Per altra banda, cada atribut té un nombre superior a 30 registres, per tant, seguint el teorema del límit central, es pot assumir que el seu estadístic de contrast es comporta com una distribució normal, per tant, es poden aplicar test per a dades normals.

Per a comprovar l'homoscedasticitat de la variància, es poden aplicar dos tipus de test: els paramètrics, en el cas que se segueixi una distribució normal i els no paramètrics en el cas de no seguir cap mena de distribució. En aquest cas, es considerarà el test de *Levene*, ja que els paramètrics solen ser més robustos i, encara que les dades no segueixin una distribució normal, d'acord amb el teorema del límit central, es considera que les dades es comporten distribució normal de mitjana de població μ i variància $\frac{\sigma^2}{\sqrt{N}}$.

```
leveneTest(y = df_titanic$Survived, group = df_titanic$Sex)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   1  5.7405 0.01678 *
##       888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(y = df_titanic$Survived, group = df_titanic$Age_group)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   2  3.3997 0.03382 *
##       887
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(y = df_titanic$Survived, group = df_titanic$Pclass)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   2   18.47 1.385e-08 ***
##       887
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(y = df_titanic$Survived, group = df_titanic$family)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   1  33.621 9.302e-09 ***
##       888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(y = df_titanic$Survived, group = df_titanic$Embarked)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   3  2.7632 0.04101 *
##       886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tal com es pot veure en les execucions anteriors del test, en tots els casos rebutgem la hipòtesi nul · la, és a dir, la hipòtesi on es considera que hi ha homoscedasticitat en la variància entre les classes comparades. Cal comentar que en aquest cas, la variable que marca la supervivència de l'individu, és una variable dicotòmica i, per tant, l'homoscedasticitat de la variància està molt relacionada amb la freqüència d'aparició.

5. Aplicació de proves estadístiques

En aquest apartat s'aplicaràn quatre tipus de proves estadístiques per a poder treure informaició de les dades i veure si així es pot predir la supervivència d'un nou individu entrenant models predictius.

Per això, es calcularà la matriu de correlacions per a tenir més informació sobre la correlació entre els atributs i també entre els atributs i la variable que marca la supervivència. A més a més, s'aplicaran contrastos d'hipòtesis entre els grups proposats anteriorment en concordança amb la matriu de correlacions.

Un cop obtinguts els resultats, s'aplicaran models per a intentar predir la supervivència dels individus. Els que s'utilitzaran són els següents:

- Regressió logistica
- Random Forest

5.1. Matriu de correlació

Per a tenir més informació entre les variables, es calcula la matriu de correlació. La informació que et proporciona és com de correlades estan les variables mitjançant el coeficient de correlació i si és una correlació positiva o negativa depenent del signe de la variable.

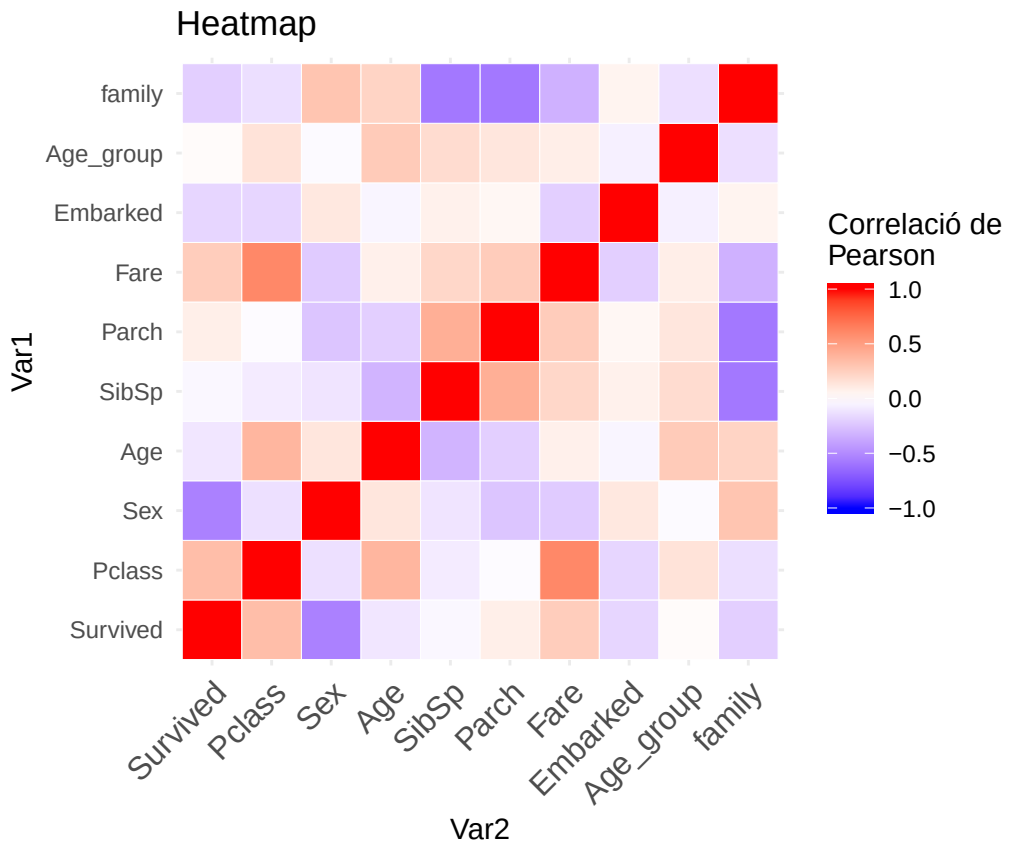
```
#aux_dataframe per a fer les variables numèriques
aux_titanic <- df_titanic
for (i in colnames(df_titanic)){
  aux_titanic[,i] <- as.numeric(aux_titanic[,i])
```

```
}
cor(aux_titanic)
```

```
##              Survived      Pclass         Sex         Age       SibSp
## Survived   1.00000000  0.33799557 -0.54305326 -0.10388818 -0.03495460
## Pclass     0.33799557  1.00000000 -0.13130534  0.37898883 -0.08271954
## Sex       -0.54305326 -0.13130534  1.00000000  0.13125416 -0.11503682
## Age       -0.10388818  0.37898883  0.13125416  1.00000000 -0.32295629
## SibSp     -0.03495460 -0.08271954 -0.11503682 -0.32295629  1.00000000
## Parch      0.08504764 -0.01551944 -0.24974553 -0.20416975  0.41577559
## Fare       0.26241156  0.60403351 -0.22090416  0.07848899  0.21101136
## Embarked  -0.17608363 -0.17306472  0.11806333 -0.04342726  0.07121504
## Age_group  0.02090061  0.14498770 -0.02095467  0.27254495  0.18335287
## family    -0.20470546 -0.13652100  0.30502030  0.22655131 -0.58443197
##                Parch        Fare    Embarked   Age_group      family
## Survived   0.08504764  0.26241156 -0.17608363  0.02090061 -0.20470546
## Pclass    -0.01551944  0.60403351 -0.17306472  0.14498770 -0.13652100
## Sex       -0.24974553 -0.22090416  0.11806333 -0.02095467  0.30502030
## Age       -0.20416975  0.07848899 -0.04342726  0.27254495  0.22655131
## SibSp      0.41577559  0.21101136  0.07121504  0.18335287 -0.58443197
## Parch      1.00000000  0.26592597  0.04145887  0.13151419 -0.58286329
## Fare       0.26592597  1.00000000 -0.20504737  0.08902935 -0.33549452
## Embarked   0.04145887 -0.20504737  1.00000000 -0.06551083  0.05687458
## Age_group  0.13151419  0.08902935 -0.06551083  1.00000000 -0.13631832
## family    -0.58286329 -0.33549452  0.05687458 -0.13631832  1.00000000
```

Per poder extreure conclusions d'una forma més còmode, es pot utilitzar l'eina de visualització *heat map*, que és un gràfic especial per a representar matrius de correlació.

```
cormat <- cor(aux_titanic)
# Transformem la matriu per a poder ser representada
melted_cormat <- melt(cormat)
# Heatmap
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
geom_tile(color = "white")+
scale_fill_gradient2(low = "blue", high = "red", mid = "white",
midpoint = 0, limit = c(-1,1), space = "Lab", name="Correlació de \nPearson") +
theme_minimal()+
theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))+
coord_fixed() + labs(title="Heatmap")
```



A més a més, es mostren els valors de més alt a més baix mitjançant el paquet *dplyr*, així es pot trobar més fàcilment els coeficients de correlació més alts amb la variable *Survived*.

```
melted_cormat <- melted_cormat %>% filter(value !=1.0) %>% filter(Var1 == "Survived")
arrange(melted_cormat, desc(abs(value)))
```

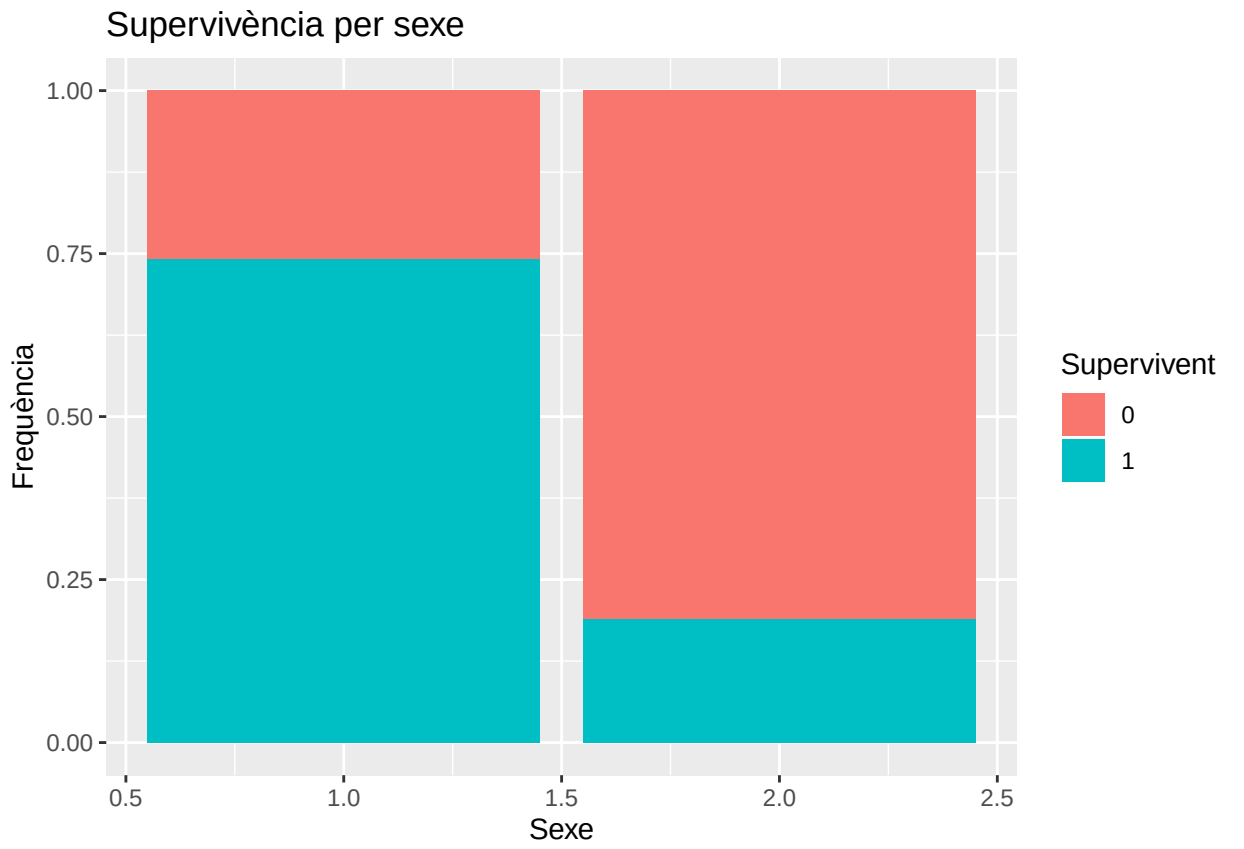
```
##       Var1      Var2       value
## 1 Survived       Sex -0.54305326
## 2 Survived    Pclass  0.33799557
## 3 Survived      Fare  0.26241156
## 4 Survived    family -0.20470546
## 5 Survived  Embarked -0.17608363
## 6 Survived       Age -0.10388818
## 7 Survived     Parch  0.08504764
## 8 Survived     SibSp -0.03495460
## 9 Survived Age_group  0.02090061
```

Tal com es pot veure, la variable que més correlacionada està amb la variable *Survived* és el sexe i la que menys és a quin grup d'edat pertany. Aquesta informació ens pot ser útil per a intentar construir models descartant les variables que menys correlació tenen amb la sortida final i, per tant, veure si el model millora utilitzant menys informació.

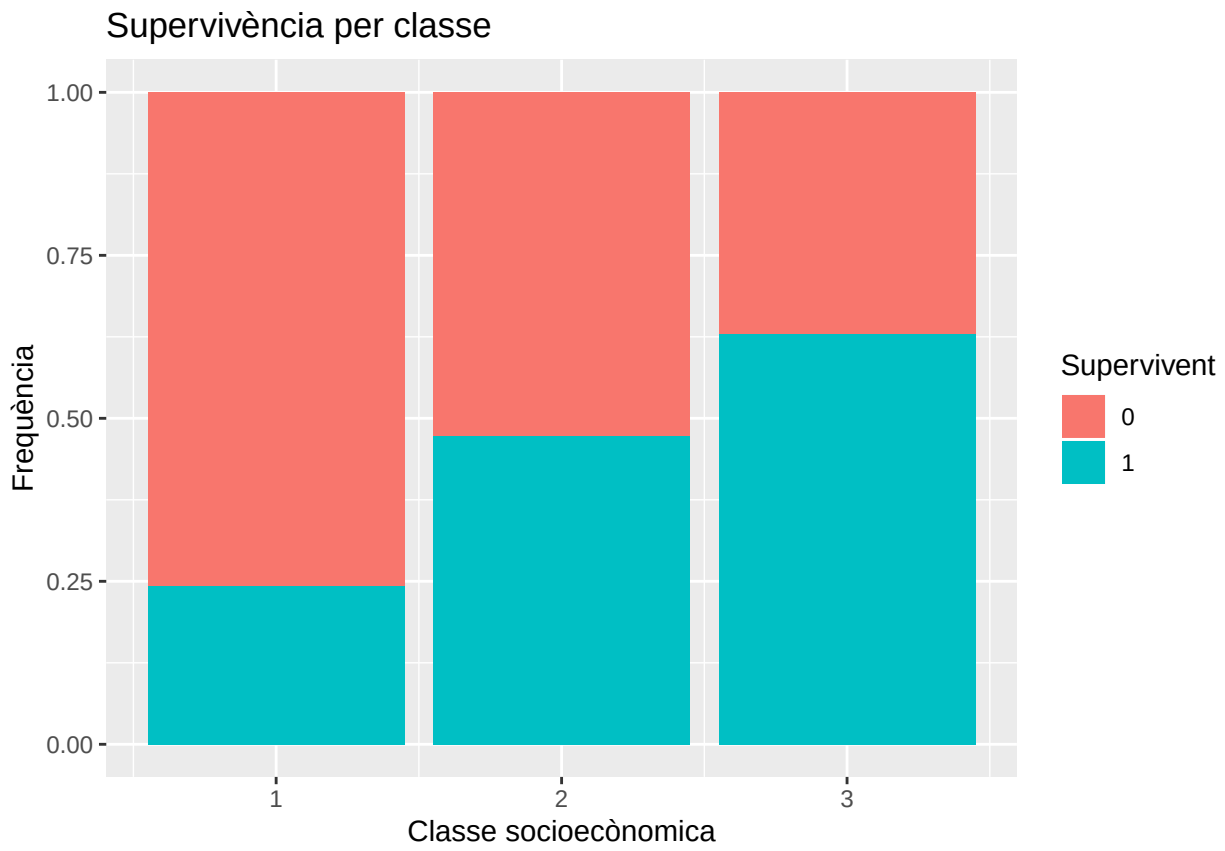
Per últim, es visualitza un diagrama de freqüència per mostrar la diferència entre la supervivència classificada pels dos atributs amb més correlació i el que menys en té.

```
ggplot(data = aux_titanic,aes(x=Sex,fill=as.factor(Survived)))+
geom_bar(position="fill")+
ylab("Frequència") +
```

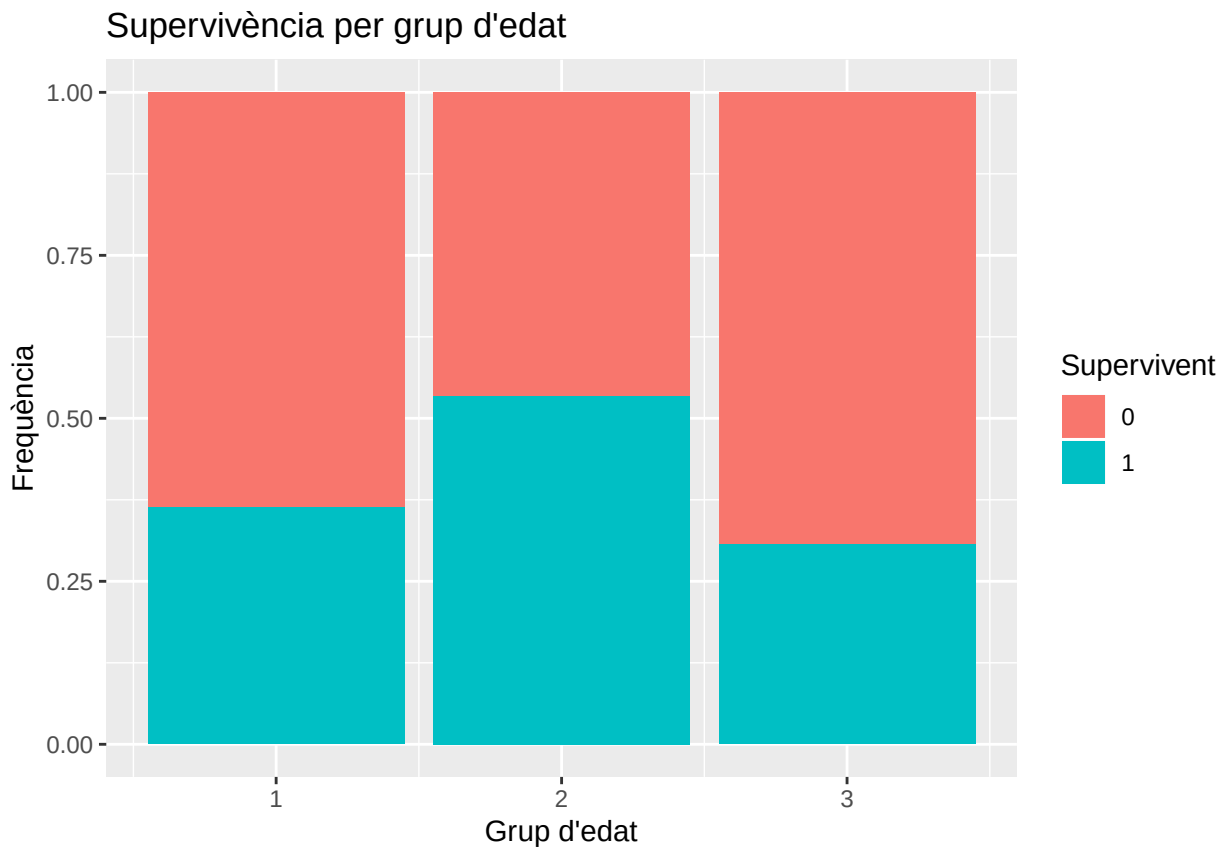
```
xlab("Sexe") +
ggtitle("Supervivència per sexe") +
labs(fill = "Supervivent")
```




```
ggplot(data = aux_titanic,aes(x=Pclass,fill=as.factor(Survived)))+
geom_bar(position="fill")+
ylab("Frequència") +
xlab("Classe socioecònomica") +
ggtitle("Supervivència per classe") +
labs(fill = "Supervivent")
```




```
ggplot(data = aux_titanic,aes(x=Age_group,fill=as.factor(Survived)))+
geom_bar(position="fill")+
ylab("Freqüència") +
xlab("Grup d'edat") +
ggtitle("Supervivència per grup d'edat") +
labs(fill = "Supervivent")
```



Com es pot veure en les gràfiques anteriors, hi ha més variablitat en la supervivència entre sexe o classe socioeconòmica que entre grups d'edat.

5.2. Contrast d'hipòtesis

Tal com s'ha vist anteriorment, la variable amb el coeficient de correlació més alt és la del sexe. Per tant, ens podem preguntar si estadísticament els homes i les dones sobreviuen per igual o hi ha una diferència significativa degut al sexe. Per a resoldre-ho, s'aplica un test de *t-student* bilateral, és a dir, les hipòtesis són les següents:

$$H_0 : \mu_0 - \mu_1 = 0$$
$$H_0 : \mu_0 - \mu_1 \neq 0$$

Seguidament, s'aplica el test per a decidir si es rebutja o s'accepta la hipòtesi nul · la.

```
t.test(df_titanic[df_titanic$Sex=="male","Survived"],
       df_titanic[df_titanic$Sex=="female","Survived"])
```

```
##
##  Welch Two Sample t-test
##
## data:  df_titanic[df_titanic$Sex == "male", "Survived"] and df_titanic[df_titanic$Sex == "female", "
## t = -18.652, df = 585.08, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6110111 -0.4945931
```

```
## sample estimates:
## mean of x mean of y
## 0.1892361 0.7420382
```

Tal com mostra el resultat del test, es pot dir que, estadísticament, amb un valor de significació del 0.05, hi ha una diferència entre les dues mitjanes i, per tant, s'ha d'acceptar que la mitjana de supervivència depèn del sexe.

Ara, s'executa el test proposant com a hipòtesi alternativa que els homes sobreviuen més que les dones.

```
t.test(df_titanic[df_titanic$Sex=="male","Survived"],
       df_titanic[df_titanic$Sex=="female","Survived"], alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  df_titanic[df_titanic$Sex == "male", "Survived"] and df_titanic[df_titanic$Sex == "female", "
## t = -18.652, df = 585.08, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.5039753
## sample estimates:
## mean of x mean of y
## 0.1892361 0.7420382
```

Tal com es mostra, també es rebutja la hipòtesi nul · la acceptant que, estadísticament, els homes de mitjana sobreviuen més que les dones.

Seguidament, es fa un contrast d'hipòtesis en la segona variable més correlacionada, la classe socioeconòmica, és a dir, es farà un contrast entre els grups establerts en l'apartat anterior de tipus de classes. Com hi ha més de dos grups, s'utilitzarà el test ANOVA per determinar si la mitjana de supervivència és estadísticament diferent en els tres tipus de classe.

```
result_aov <- aov(Survived ~ Pclass, data = df_titanic)
summary(result_aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Pclass        2  24.25   12.12   57.71 <2e-16 ***
## Residuals   887 186.33    0.21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tal com marca el resultat anterior, es pot dir que, estadísticament, hi ha diferència entre les mitjanes de supervivència de les diferents classe.

Per altra banda, tal com s'ha fet en l'apartat anterior, es farà un test ANOVA per a la variable menys correlada, és a dir, per al grup d'edat.

```
result_aov <- aov(Survived ~ Age_group, data = df_titanic)
summary(result_aov)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## Age_group     2   3.76  1.8798   8.062 0.000339 ***
## Residuals   887 206.82  0.2332
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En aquest cas, encara que la correlació sigui baixa, no es pot afirmar que estadísticament les mitjanes de les variables siguin iguals.

5.3. Regressió logistica

En aquest apartat, s'intentarà construir, utilitzant la informació obtinguda en els apartats anteriors, un model que sigui capaç de classificar entre si un individu sobreviu o no sobreviu. Primerament, s'utilitzarà un model logístic amb totes les variables que es tenen al *dataset*. Per evitar *overfitting*, s'utilitzarà la validació creuada per extreure la capacitat de predir del model.

```
# Utilitzem el dataframe recodificat a numeric anteriorment
# Reconfigurem la variable Survived com a factor per a fer classificació
aux_titanic$Survived[aux_titanic$Survived==1] = "yes"
aux_titanic$Survived[aux_titanic$Survived==0] = "no"
aux_titanic$Survived <- as.factor(aux_titanic$Survived)
complete_log_model <- glm(Survived ~ ., data = aux_titanic, family = binomial)
summary(complete_log_model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = aux_titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7539  -0.6143  -0.3626   0.5908   2.5973
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.070081   0.753287   6.731 1.69e-11 ***
## Pclass       1.302302   0.162398   8.019 1.06e-15 ***
## Sex         -2.653409   0.203656 -13.029  < 2e-16 ***
## Age         -0.054075   0.008366  -6.463 1.02e-10 ***
## SibSp       -0.624868   0.154158  -4.053 5.05e-05 ***
## Parch       -0.216244   0.143434  -1.508   0.1317
## Fare        -0.002364   0.003089  -0.765   0.4441
## Embarked    -0.187803   0.117517  -1.598   0.1100
## Age_group    0.213807   0.151925   1.407   0.1593
## family      -0.706874   0.286053  -2.471   0.0135 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1185.7  on 889  degrees of freedom
## Residual deviance:  757.3  on 880  degrees of freedom
## AIC: 777.3
##
## Number of Fisher Scoring iterations: 5
```

En aquest cas, es pot veure que segons la columna que marca el valor p de si la variable sigui pertanyent a aquest model, amb un nivell de significació de 0.05, les variables que pertanyen al model són: *Pclass*, *Sex*, *Age* i *SibSp*.

A continuació, s'utilitzarà la validació creuada per a veure el percentatge d'encert del model construït. A més a més, s'utilitzaran les variables esmentades i les variables amb un coeficient de correlació superior a 0.10 en valor absolut vistes en l'apartat 5.1 per a crear dos models més per a comparar amb el primer. Els models creats seran anomenats model complet, model versió 1 i model versió 2, respectivament.

```r
#separem dades d'entrenament i dades de avaluació
h<-holdout(aux_titanic$Survived,ratio=2/3,mode="stratified")
data_train<-aux_titanic[h$tr,]
data_test<-aux_titanic[h$ts,]

# complete
h<-holdout(aux_titanic$Survived,ratio=2/3,mode="stratified")
data_train<-aux_titanic[h$tr,]
data_test<-aux_titanic[h$ts,]
train_control<- trainControl(method="cv", number=10)
mod<-train(Survived~., data=data_train, method="glm", trControl = train_control)
pred <- predict(mod, newdata=data_test)
completelg <- confusionMatrix(pred, data_test$Survived, positive = "yes")
completelg
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  155  25
##        yes  28  89
##
##                Accuracy : 0.8215
##                  95% CI : (0.7732, 0.8634)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : 1.281e-14
##
##                   Kappa : 0.6246
##
##  Mcnemar's Test P-Value : 0.7835
##
##             Sensitivity : 0.7807
##             Specificity : 0.8470
##          Pos Pred Value : 0.7607
##          Neg Pred Value : 0.8611
##              Prevalence : 0.3838
##          Detection Rate : 0.2997
##    Detection Prevalence : 0.3939
##       Balanced Accuracy : 0.8138
##
##        'Positive' Class : yes
##
```

```r
# V1
train_control<- trainControl(method="cv", number=10)
mod<-train(Survived ~ Pclass + Sex + Age + SibSp + family,
           data=data_train, method="glm", trControl = train_control)
pred <- predict(mod, newdata=data_test)
v1lg <- confusionMatrix(pred, data_test$Survived, positive = "yes")
v1lg
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction  no yes
##        no  154  26
##        yes  29  88
##
##                Accuracy : 0.8148
##                  95% CI : (0.7659, 0.8573)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : 1.017e-13
##
##                   Kappa : 0.6104
##
##  Mcnemar's Test P-Value : 0.7874
##
##             Sensitivity : 0.7719
##             Specificity : 0.8415
##          Pos Pred Value : 0.7521
##          Neg Pred Value : 0.8556
##              Prevalence : 0.3838
##          Detection Rate : 0.2963
##    Detection Prevalence : 0.3939
##       Balanced Accuracy : 0.8067
##
##        'Positive' Class : yes
##
```

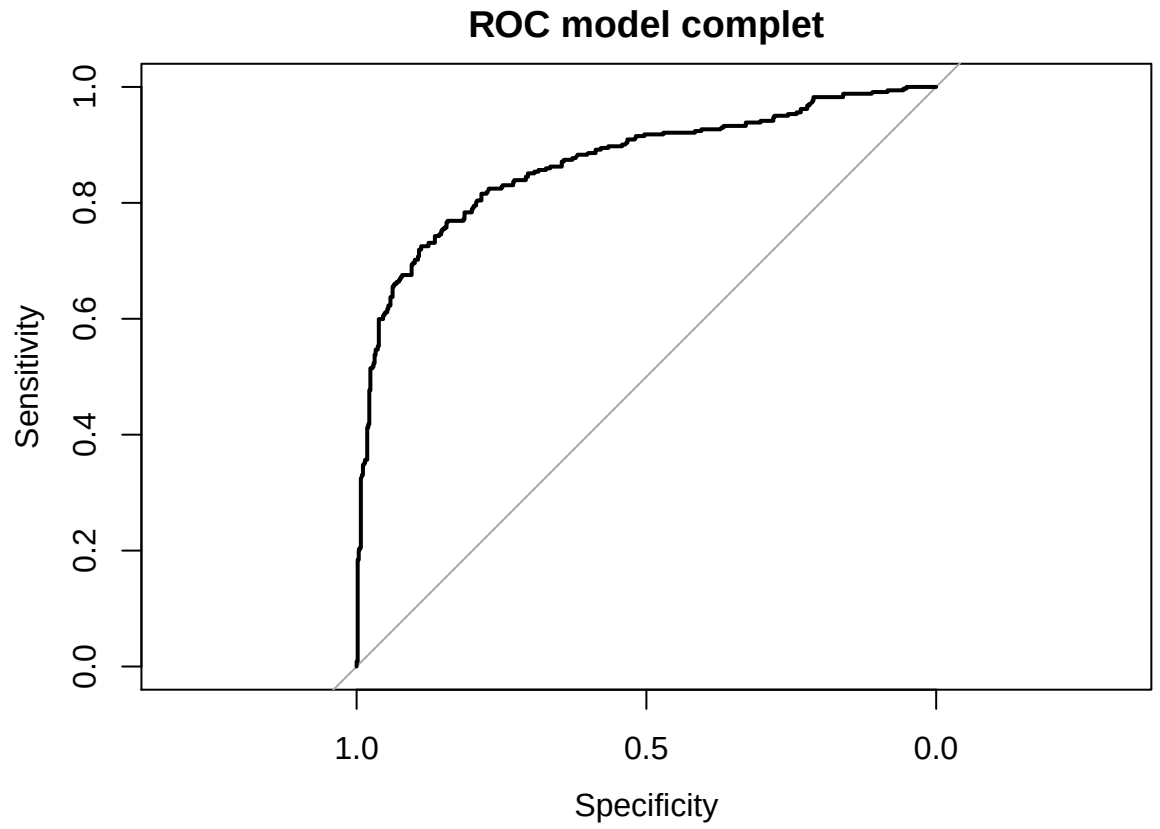
```
# V2
train_control<- trainControl(method="cv", number=10)
mod<-train(Survived ~ Sex + Pclass + Fare + family + Embarked + Age,
           data=data_train, method="glm", trControl = train_control)
pred <- predict(mod, newdata=data_test)
v2lg<-confusionMatrix(pred, data_test$Survived, positive = "yes")
v2lg
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  154  27
##        yes  29  87
##
##                Accuracy : 0.8114
##                  95% CI : (0.7622, 0.8543)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : 2.772e-13
##
##                   Kappa : 0.6027
##
##  Mcnemar's Test P-Value : 0.8937
##
##             Sensitivity : 0.7632
##             Specificity : 0.8415
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.8508
##              Prevalence : 0.3838
##          Detection Rate : 0.2929
```

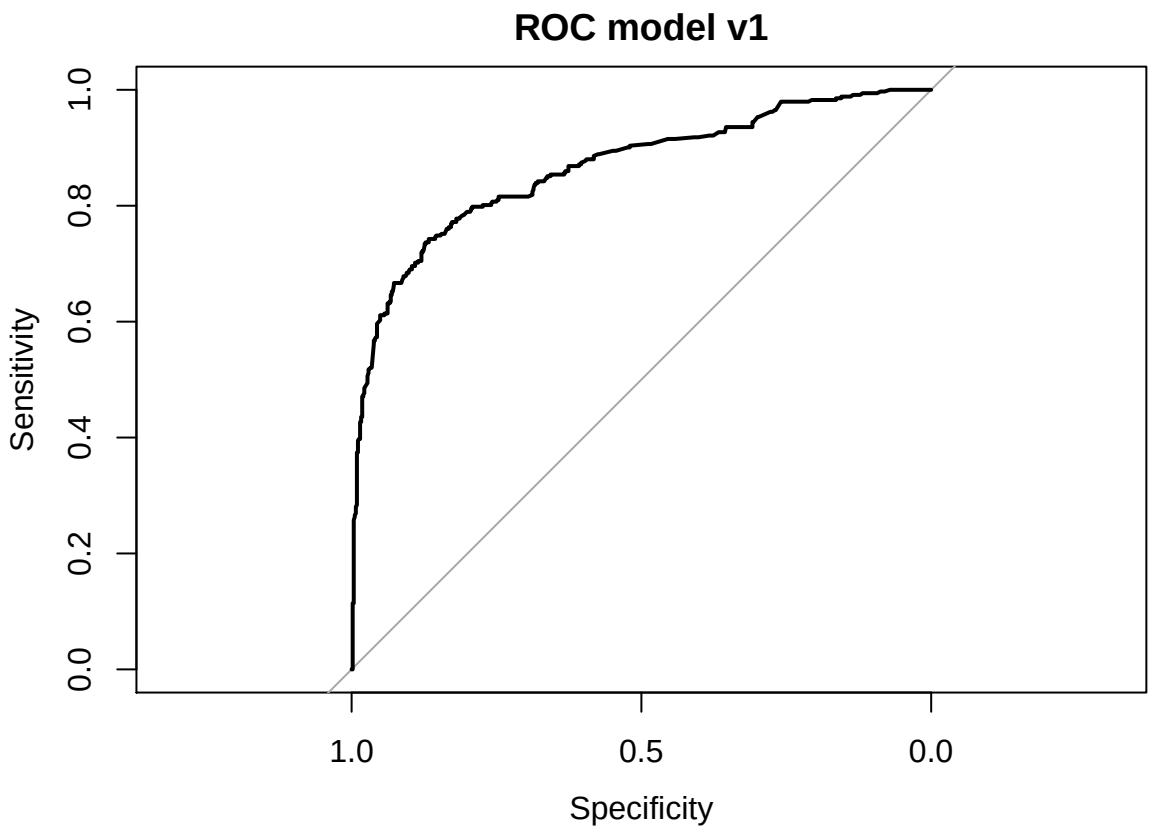
```
##    Detection Prevalence : 0.3906
##       Balanced Accuracy : 0.8023
##
##        'Positive' Class : yes
##
```

Si s'examinen els resultats anteriors, els tres models presenten un percentatge d'encerts molt similar, al voltant de 80%. Per tant, no es pot dir que hi hagi un model millor que l'altre i, per tant, s'utilitzarà la corba ROC i l'àrea sota la corba per veure si es pot decidir quin model és millor que l'altre en aquest cas.

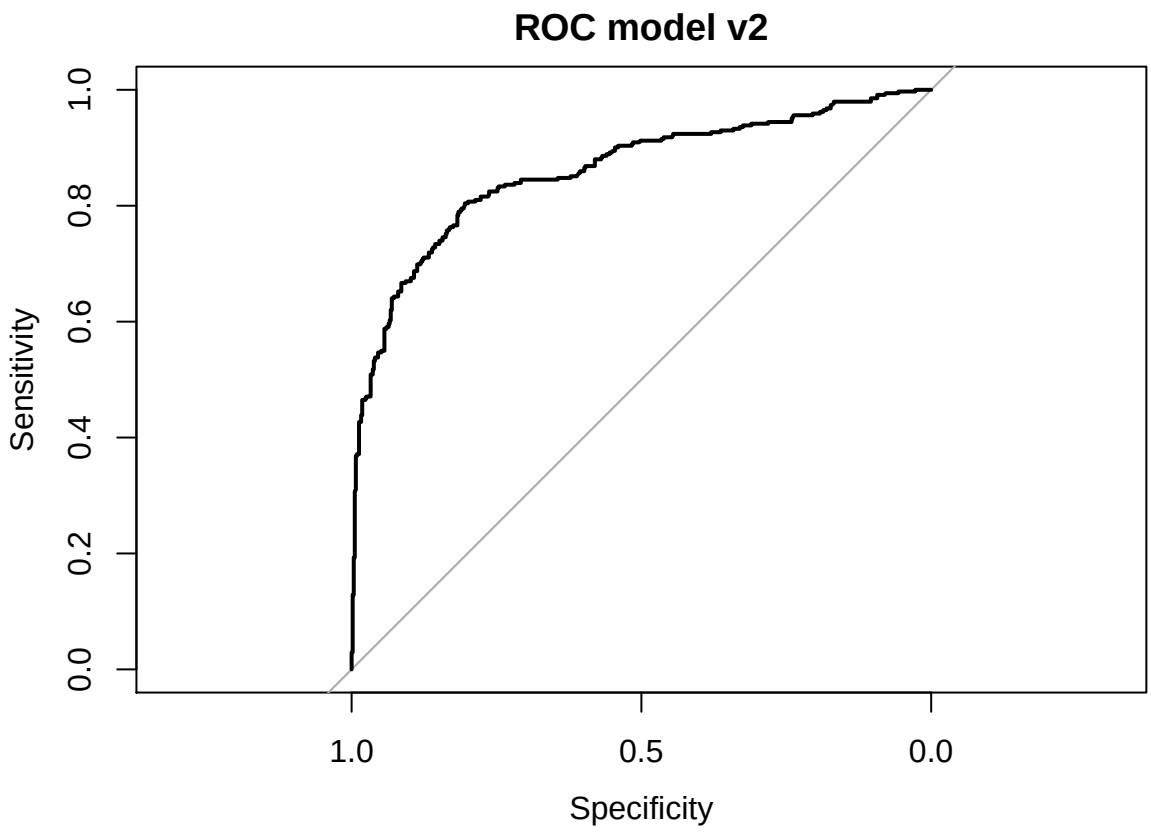
```
# roc
dat_aux <- aux_titanic
log_model_v2 <- glm(Survived ~ Sex + Pclass + Fare + family + Embarked + Age,
                                              data = aux_titanic, family = binomial)
prob=predict(complete_log_model,type=c("response"))
dat_aux$prob=prob
roc <- roc(Survived ~ prob, data = dat_aux)
plot(roc, main="ROC model complet")
```




```
# roc
dat_aux <- aux_titanic
log_model_v1 <- glm(Survived ~ Pclass + Sex + Age + SibSp,
                    data = aux_titanic, family = binomial)
prob=predict(log_model_v1,type=c("response"))
dat_aux$prob=prob
roc1 <- roc(Survived ~ prob, data = dat_aux)
plot(roc1, main="ROC model v1")
```




```
# roc
dat_aux <- aux_titanic
log_model_v2 <- glm(Survived ~ Sex + Pclass + Fare + family + Embarked + Age,
                                          data = aux_titanic, family = binomial)
prob=predict(log_model_v2,type=c("response"))
dat_aux$prob=prob
roc2 <- roc(Survived ~ prob, data = dat_aux)
plot(roc2, main="ROC model v2")
```




```
auc_completelg <- auc(roc)
auc_completelg
```

```
## Area under the curve: 0.8689
```

```
auc_v1lg<-auc(roc1)
auc_v1lg
```

```
## Area under the curve: 0.865
```

```
auc_v2lg<-auc(roc2)
auc_v2lg
```

```
## Area under the curve: 0.86
```

Tal com es veu en les execucions anteriors, en la visualització de la corba ROC no es veu cap diferència significativa de quin model és millor, a més a més, cal comentar que el valor de l'àrea sota la corba és molt similar en tots tres casos, ratificant el que s'ha vist en les dues proves anteriors.

Per últim, es mostra una taula comparativa de les dades recollides anteriorment.

```
complet_log <- c(round(as.numeric(completelg$overall["Accuracy"]), 2),
                 round(as.numeric(completelg$byClass["Sensitivity"]), 2),
                 round(as.numeric(completelg$byClass["Specificity"]), 2),
                 round(as.numeric(auc_completelg), 2))
v1_log <- c(round(as.numeric(v1lg$overall["Accuracy"]), 2),
            round(as.numeric(v1lg$byClass["Sensitivity"]), 2),
            round(as.numeric(v1lg$byClass["Specificity"]), 2),
            round(as.numeric(auc_v1lg), 2))
```

```
v2_log <- c(round(as.numeric(v2lg$overall["Accuracy"]), 2),
            round(as.numeric(v2lg$byClass["Sensitivity"]), 2),
            round(as.numeric(v2lg$byClass["Specificity"]), 2),
            round(as.numeric(auc_v2lg), 2))

result_table <- data.frame(rbind(complet_log, v1_log, v2_log))
names(result_table) <- c("Precisió","Sensibilitat","Specificitat","AUC")
result_table
```

```
##             Precisió Sensibilitat Specificitat  AUC
## complet_log     0.82         0.78         0.85 0.87
## v1_log          0.81         0.77         0.84 0.86
## v2_log          0.81         0.76         0.84 0.86
```

Com a conclusió dels models logístics, es pot dir que no hi ha diferència entre models i, per tant, es pot utilitzar el model amb menys variables, ja que per jocs de dades molt grans pot suposar un augment en la rapidesa d'aquest en predir.

5.4. Random forest

En aquest apartat, s'entrenarà un model basat en l'algoritme *random forest* per a fer una predicció de quins són els individus que sobreviuen. Per a poder avaluar-lo, s'utilitzarà la tècnica de *cross-validation* igual que en l'apartat anterior.

A més a més, també s'entrenaran tres models fixant-nos en els criteris establerts anteriorment per a veure si utilitzant un model basat en *random forest* hi ha diferències entre aquests.

```
# complete
data_train<-aux_titanic[h$tr,]
data_test<-aux_titanic[h$ts,]
train_control<- trainControl(method="cv", number=10)
mod<-train(Survived~., data=data_train, method="rf", trControl = train_control)
pred <- predict(mod, newdata=data_test)
completerf<-confusionMatrix(pred, data_test$Survived, positive = "yes")
completerf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  161  24
##        yes  22  90
##
##                Accuracy : 0.8451
##                  95% CI : (0.7989, 0.8843)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6715
##
##  Mcnemar's Test P-Value : 0.8828
##
##             Sensitivity : 0.7895
##             Specificity : 0.8798
##          Pos Pred Value : 0.8036
##          Neg Pred Value : 0.8703
```

```
##              Prevalence : 0.3838
##          Detection Rate : 0.3030
##    Detection Prevalence : 0.3771
##       Balanced Accuracy : 0.8346
##
##        'Positive' Class : yes
##
```

```
# v1
train_control<- trainControl(method="cv", number=10)
mod<-train(Survived ~ Pclass + Sex + Age + SibSp + family, data=data_train,
           method="rf", trControl = train_control)
pred <- predict(mod, newdata=data_test)
v1rf<-confusionMatrix(pred, data_test$Survived, positive = "yes")
v1rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  164  30
##        yes  19  84
##
##                Accuracy : 0.835
##                  95% CI : (0.7878, 0.8754)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6447
##
##  Mcnemar's Test P-Value : 0.1531
##
##             Sensitivity : 0.7368
##             Specificity : 0.8962
##          Pos Pred Value : 0.8155
##          Neg Pred Value : 0.8454
##              Prevalence : 0.3838
##          Detection Rate : 0.2828
##    Detection Prevalence : 0.3468
##       Balanced Accuracy : 0.8165
##
##        'Positive' Class : yes
##
```

```
# v2
train_control<- trainControl(method="cv", number=10)
mod<-train(Survived ~ Sex + Pclass + Fare + family + Embarked + Age,
           data=data_train, method="rf", trControl = train_control)
pred <- predict(mod, newdata=data_test)
v2rf<-confusionMatrix(pred, data_test$Survived, positive = "yes")
v2rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
```

```
##        no  168  29
##        yes  15  85
##
##                Accuracy : 0.8519
##                  95% CI : (0.8063, 0.8902)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6794
##
##  Mcnemar's Test P-Value : 0.05002
##
##             Sensitivity : 0.7456
##             Specificity : 0.9180
##          Pos Pred Value : 0.8500
##          Neg Pred Value : 0.8528
##              Prevalence : 0.3838
##          Detection Rate : 0.2862
##    Detection Prevalence : 0.3367
##       Balanced Accuracy : 0.8318
##
##        'Positive' Class : yes
##
```

Si es comparen els resultats entre models, es pot veure que, igual que en el cas del model de regressió logística, no hi ha una gran diferència entre els resultats dels models, sent la precisió una mica més alta que en el cas anterior, al voltant d'un 80%, però sense ser significativament més alta que en els de regressió. Per tant, no es pot afirmar que aquests models basats en *random forest* siguin millor que els anteriors.

Per últim, es mostra una taula comparativa de les dades recollides.

```
complet_rf <- c(round(as.numeric(completerf$overall["Accuracy"]), 2),
                round(as.numeric(completerf$byClass["Sensitivity"]), 2),
                round(as.numeric(completerf$byClass["Specificity"]), 2))
v1_rf <- c(round(as.numeric(v1rf$overall["Accuracy"]), 2),
           round(as.numeric(v1rf$byClass["Sensitivity"]), 2),
           round(as.numeric(v1rf$byClass["Specificity"]), 2))
v2_rf <- c(round(as.numeric(v2rf$overall["Accuracy"]), 2),
           round(as.numeric(v2rf$byClass["Sensitivity"]), 2),
           round(as.numeric(v2rf$byClass["Specificity"]), 2))

result_table <- data.frame(rbind(complet_rf, v1_rf, v2_rf))
names(result_table) <- c("Precisió","Sensibilitat","Especificitat")
result_table
```

```
##            Precisió Sensibilitat Especificitat
## complet_rf     0.85         0.79          0.88
## v1_rf          0.84         0.74          0.90
## v2_rf          0.85         0.75          0.92
```

6. Presentació de resultats

Un cop avaluat els diferents models, es pot mostrar els resultats de tots ells, a més a més, fer una comparativa per a veure les seves similituds i diferències. Primerament, es mostra en una taula i en forma de diagrama de barres de les mètriques de precisió, sensibilitat i especificitat dels diferents models.

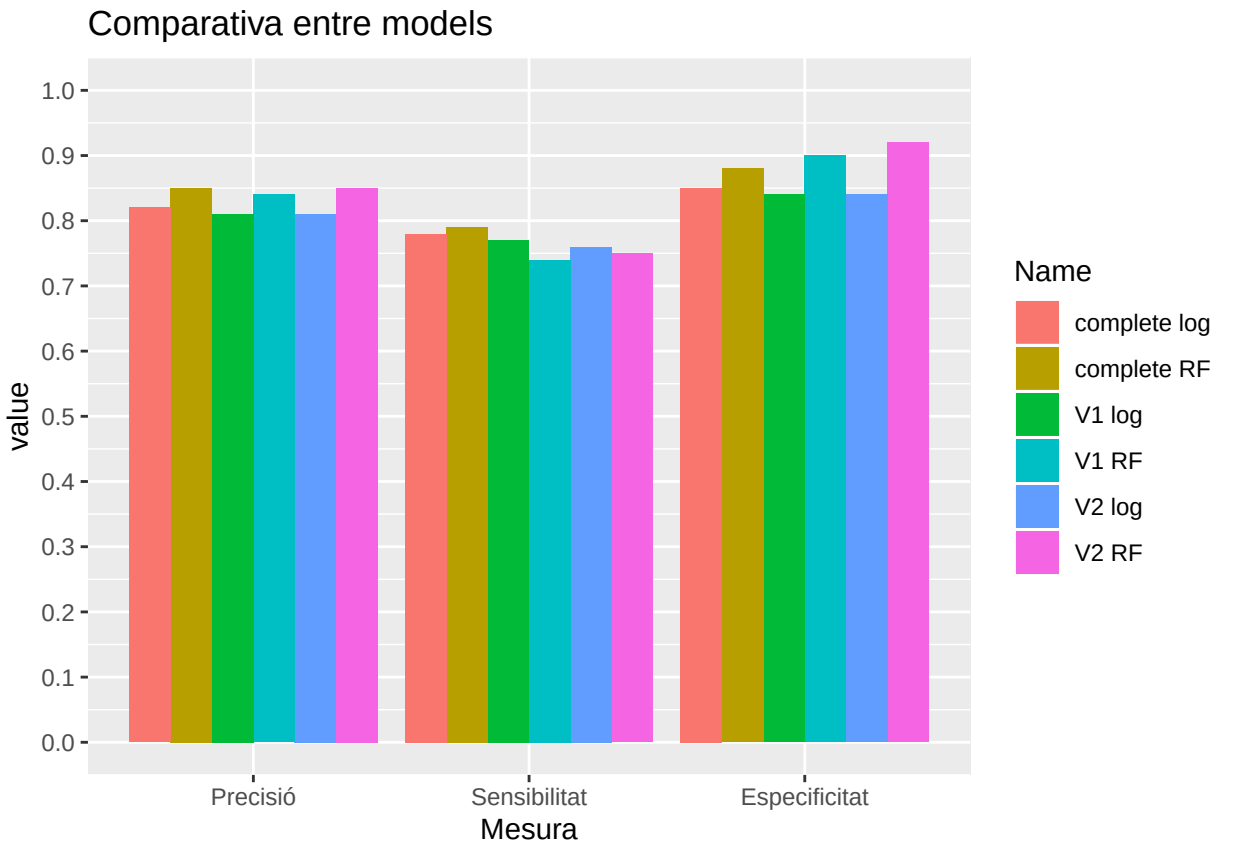
```
complet_rf <- c("complete RF",
                round(as.numeric(completerf$overall["Accuracy"]), 2),
                round(as.numeric(completerf$byClass["Sensitivity"]), 2),
                round(as.numeric(completerf$byClass["Specificity"]), 2))
v1_rf <- c("V1 RF",
           round(as.numeric(v1rf$overall["Accuracy"]), 2),
           round(as.numeric(v1rf$byClass["Sensitivity"]), 2),
           round(as.numeric(v1rf$byClass["Specificity"]), 2))
v2_rf <- c("V2 RF",
           round(as.numeric(v2rf$overall["Accuracy"]), 2),
           round(as.numeric(v2rf$byClass["Sensitivity"]), 2),
           round(as.numeric(v2rf$byClass["Specificity"]), 2))
complet_log <- c("complete log",
                 round(as.numeric(completelg$overall["Accuracy"]), 2),
                 round(as.numeric(completelg$byClass["Sensitivity"]), 2),
                 round(as.numeric(completelg$byClass["Specificity"]), 2))
v1_log <- c("V1 log",
            round(as.numeric(v1lg$overall["Accuracy"]), 2),
            round(as.numeric(v1lg$byClass["Sensitivity"]), 2),
            round(as.numeric(v1lg$byClass["Specificity"]), 2))
v2_log <- c("V2 log",
            round(as.numeric(v2lg$overall["Accuracy"]), 2),
            round(as.numeric(v2lg$byClass["Sensitivity"]), 2),
            round(as.numeric(v2lg$byClass["Specificity"]), 2))

result_table <- data.frame(rbind(complet_rf, v1_rf, v2_rf, complet_log, v1_log, v2_log))
names(result_table) <- c("Name","Precisió","Sensibilitat","Especificitat")
result_table
```

```
##                      Name Precisió Sensibilitat Especificitat
## complet_rf   complete RF     0.85         0.79          0.88
## v1_rf              V1 RF     0.84         0.74           0.9
## v2_rf              V2 RF     0.85         0.75          0.92
## complet_log complete log     0.82         0.78          0.85
## v1_log            V1 log     0.81         0.77          0.84
## v2_log            V2 log     0.81         0.76          0.84
```

```
mdat <- melt(result_table, id.vars="Name")
mdat$value <- round(as.numeric(mdat$value),2)

ggplot(mdat, aes(variable, value, fill=Name)) +
geom_bar(stat="identity", position="dodge") +
scale_y_continuous(breaks = seq(0, 1, by=0.1), limits= c(0,1)) +
xlab("Mesura") +
ggtitle("Comparativa entre models")
```



Si s'observa la taula mostrada, es pot veure que els models basats en els algorismes *random forest* estan alguns punts per sobre en el percentatge de totes les mesures. Cal dir que aquests són algoritmes més complexos i, per tant, més lents d'executar. A més a més, si es mira mètrica per mètrica, es pot veure que els models construïts basants en el *random forest* amb totes les dades i amb la versió 2, és a dir, la que prioritza les variables amb coeficient de correlació superior a 0.10, tenen, en general, un percentatge més alt en precisió, sensibilitat i especificitat.

Com que tots els models tenen valors similars en les mètriques, la seva elecció vindrà determinada per la mesura que vulguem prioritzar, és a dir, si es vol un algorisme més senzill amb bona sensibilitat, es pot escollir el de regressió logística amb versió 2, així es tindrà rapidesa, menys processament de dades i bona sensibilitat. En canvi, si es vol la màxima precisió, s'ha d'utilitzar el model complet_RF o V2_rf, sent el segon més lleuger, ja que processa menys dades.

7. Conclusions

L'objectiu d'aquest estudi era poder predir si hi ha un individu havia sobreviscut a l'accident del Titànic tenint en compte els paràmetres: sexe, edat, nombre de germans, marits i mullers a bord, nombre de fills i pares a bord, número del tiquet, tarifa del passatger, número de cabina i port de l'embarcació.

Primerament, s'ha fet un anàlisi per trobar anomalies en les dades, com per exemple valors extrems, seguit d'una neteja de dades, un analísi de dades i, finalment, una construcció de diversos models basats en dos tipus diferents d'algorisme per intentar predir si els individus sobreviurien.

Finalment, després de la construcció del model utilitzant les dades processades en apartats anteriors, es pot concloure que amb una precisió al voltant del 80% d'encert, sí que es pot predir si un individu sobreviuria o no ho farà utilitzant els paràmetres donats.

8. Referències

1. Dalgaard, P. (2002). Introductory statistics with R. Springer Science & Business Media.
2. Calvo, M.; Pérez, D.O.; Subirats,L. (2019). Introducció a la neteja i anàlisi de dades. Material UOC.
3. Test for homogeneity of variances - Lavene's test and the Fligner Killeen test (2016) [en línea]. bioSt@TS. [Consulta: 26 de diciembre de 2017] https://biostats.w.uib.no/ test-for-homogeneity-of-variances-levenes-test/