

Tipologia i Cicle de vida de les dades

Pràctica 2: Neteja i validació de les dades

Pol Casellas, Carlos Rivas

10 de June, 2019

- 1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre?
- 2. Integració i selecció de les dades d'interès a analitzar.
- 3. Neteja de les dades.
 - 3.1 Les dades contenen zeros o elements buits? Com gestionar aquests casos?
 - 3.2. Identificació i tractament de valors extrems.
- 4. Anàlisi de les dades.
 - 4.1. Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar).
 - 4.2. Comprovació de la normalitat i homogeneïtat de la variància.
 - 4.3. Aplicació de proves estadístiques per comparar els grups de dades. En funció de les dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesis, correlacions, regressions, etc. Aplicar almenys tres mètodes d'anàlisi diferents.
 - Hipòtesi nul·la i alternativa
 - Assumpció de normalitat
 - Homogeneïtat de la variància
 - Correlació
 - Regressió logística
- 5. Representació dels resultats a partir de taules i gràfiques.
- 6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema?
 - 6.1.- Procés de neteja i validació
 - 6.2.- Anàlisi de dades

1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre?

El dataset original és el dataset de dades dels passatgers del Titànic, que s'utilitza en la competició "Titanic: Machine Learning from Disaster". Esta compostat per dos fitxers, un de training (train.csv) amb 891 registres i un de test (test.csv) amb 418 registres. L'objectiu del dataset de la competició es generar un model de ML per predir la supervivència a l'enfonsament del vaixell, en base a la informació disponible dels passatgers.

El diccionari de dades del dataset original és :

Variable	Descripció	Tipus	Valors
PassengerId	Identificador de passatger	Enter	
Survival	Identificador de supervivència	Factor	0 = No, 1 = Yes
Pclass	Classe del passatge	Factor	1 = 1a, 2 = 2ona, 3 = 3 ^a
Name	Nom del passatger	Text	
Sex	Sexe	Factor	1 = male, 2 = female
Age	Edat en anys	Nombre	
Sibsp	Nombre de germans i dones embarcats	Enter	
Parch	Nombre de pares i fills embarcats	Enter	
Ticket	Nombre de Ticket	Text	
Fare	Tarifa del passatge	Nombre	
Cabin	Nombre de cabina	Text	
Embarked	Port d'Embarcament	Factor	C = Cherbourg Q = Queenstown S = Southampton

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

Es disposa de 2 fitxers, un fitxer d'entrenament (que disposa de les dades de supervivència) i un fitxer de test (sense les dades de supervivència). Es llegeixen les dades i s'integren en un únic dataset.

```
library('dplyr')
dades <- bind_rows(read.csv('train.csv', stringsAsFactors = F), read.csv('test.csv', stringsAsFactors = F))
```

Es crea un dataset que integra els dos fitxers anomenat `dades`. La comanda `dim(dades)` mostra el nombre d'observacions i atributs de la mostra, que en aquest cas són 1309 observacions i 12 atributs. El camp **Survived** només està informat per les entrades del fitxer d'entrenament (fins la posició 891). En la resta d'entrades, el valor en N.A.

```
dim (dades)
```

```
## [1] 1309 12
```

La comanda `str(dades)` presenta de forma compacta l'estructura interna del dataset.

```
str(dades)
```

```
## 'data.frame': 1309 obs. of 12 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
## $ Sex : chr "male" "female" "female" "female" ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr "" "C85" "" "C123" ...
## $ Embarked : chr "S" "C" "S" "S" ...
```

Amb la comanda

```
factors<-c('Sex','Embarked')
dades[factors] <- lapply(dades[factors], function(x) as.factor(x))
```

s'indica al dataset que els camps **Pclass**, **Sex** i **Embarked** són factors. Al aplicar de nou la funció `str(dades)` es pot observar que ara es consideren factors i es mostren els possibles valors que admeten.

```
str(dades)
```

```
## 'data.frame': 1309 obs. of 12 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
## $ 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 : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr "" "C85" "" "C123" ...
## $ Embarked : Factor w/ 4 levels "", "C", "Q", "S": 4 2 4 4 4 3 4 4 4 2 ...
```

La comanda `summary(dades)` mostra una visió general dels atributs i en el cas de factors, es presenten els valors possibles i el nombre d'ocurrències per cada valor. A més, es pot observar que hi ha diversos valors mal informats o buits.

```
summary (dades)
```

```
## PassengerId Survived Pclass Name
## Min. : 1 Min. :0.0000 Min. :1.000 Length:1309
## 1st Qu.: 328 1st Qu.:0.0000 1st Qu.:2.000 Class :character
## Median : 655 Median :0.0000 Median :3.000 Mode :character
## Mean : 655 Mean :0.3838 Mean :2.295
## 3rd Qu.: 982 3rd Qu.:1.0000 3rd Qu.:3.000
## Max. :1309 Max. :1.0000 Max. :3.000
## NA's :418
## Sex Age SibSp Parch
## female:466 Min. : 0.17 Min. :0.0000 Min. :0.000
## male :843 1st Qu.:21.00 1st Qu.:0.0000 1st Qu.:0.000
## Median :28.00 Median :0.0000 Median :0.000
## Mean :29.88 Mean :0.4989 Mean :0.385
## 3rd Qu.:39.00 3rd Qu.:1.0000 3rd Qu.:0.000
## Max. :80.00 Max. :8.0000 Max. :9.000
## NA's :263
## Ticket Fare Cabin Embarked
## Length:1309 Min. : 0.000 Length:1309 : 2
## Class :character 1st Qu.: 7.896 Class :character C:270
## Mode :character Median :14.454 Mode :character Q:123
## Mean :33.295 S:914
## 3rd Qu.:31.275
## Max. :512.329
## NA's :1
```

3. Neteja de les dades.

3.1 Les dades contenen zeros o elements buits? Com gestionaries aquests casos?

Es pot observar que hi ha 418 valors no informats de Survived que corresponen als valors del fitxer de test, 263 registres no presents a **Age**, 1 registre que no informa el **Fare** i 2 registres que no indiquen el camp Embarked. Addicionalment hi ha diversos registres de **Fare** amb valor 0. Es procedeix a aplicar diverses tècniques per completar aquestes dades.

- Per intentar determinar el valor de la tarifa (**Fare**) que manca, el primer que es fa és identificar el registre mal informat

```
IdNoFare <- dades$PassengerId[is.na(dades$Fare)]
dades[IdNoFare,]
```

```
## PassengerId Survived Pclass Name Sex Age SibSp Parch
## 1044 1044 NA 3 Storey, Mr. Thomas male 60.5 0 0
## Ticket Fare Cabin Embarked
## 1044 3701 NA S
```

S'observa que es tracta d'un passatger de classe 3 i que va embarcar a 'S'. Es calcula el preu mitjà de la tarifa d'aquest tipus de clients i s'assigna a la tarifa del passatger no informat. El càlcul de la mitjana no té en compte els valors NA (paràmetre na.rm = TRUE)

```
dades$Fare[is.na(dades$Fare)] <- median(dades[dades$Pclass == '3' & dades$Embarked == 'S', ]$Fare, na.rm = TRUE)
dades[IdNoFare,]
```

```
## PassengerId Survived Pclass Name Sex Age SibSp Parch
## 1044 1044 NA 3 Storey, Mr. Thomas male 60.5 0 0
## Ticket Fare Cabin Embarked
## 1044 3701 8.05 S
```

El valor assignat és 8,05 \$

- Per intentar determinar el valor de l'Embarked que manca, el primer que es fa és identificar els registres mal informats

```
IdNoEmbarked <- dades$PassengerId[dades$Embarked == '']
dades[IdNoEmbarked,]
```

```
##      PassengerId Survived Pclass                                Name
## 62             62         1      1                                Icard, Miss. Amelie
## 830            830         1      1 Stone, Mrs. George Nelson (Martha Evelyn)
##      Sex Age SibSp Parch Ticket Fare Cabin Embarked
## 62  female  38      0      0 113572   80   B28
## 830 female  62      0      0 113572   80   B28
```

S'observa que es tracta de passatgers de classe 1 amb una tarifa de 80 \$. Per determinar el port d'embarcament es calcula la tarifa mitjana per a la classe 1, per cada tipus d'embarcament possible.

```
median(dades[daes$Pclass == '1' & daes$Embarked == 'S', ]$Fare, na.rm = TRUE)
```

```
## [1] 52
```

```
median(dades[daes$Pclass == '1' & daes$Embarked == 'Q', ]$Fare, na.rm = TRUE)
```

```
## [1] 90
```

```
median(dades[daes$Pclass == '1' & daes$Embarked == 'C', ]$Fare, na.rm = TRUE)
```

```
## [1] 76.7292
```

S'observa que la tarifa mitjana més propera a la dels registres no informats és 76,73 \$ corresponent a l'embarcament 'C'. S'assigna aquest valor als registres mal informats.

```
daes$Embarked[c(IdNoEmbarked)] <- 'C'
```

- Donat que el nombre de valors **Age** no informats és elevat, la opció d'esborrar aquests valors suposaria la pèrdua de molts registres que si contenen altres valors i que és interessant conservar. Per aquest motiu, s'utilitzarà la llibreria *mice* (Multivariate Imputations by Chained Equations) que implementa un mètode per tractar valors no informats, creant múltiples imputacions per dades no informades multivariable. Cada variable incompleta es pot imputar per un model separat, permetent la imputació de barreges de dades de diversos tipus (categòriques, contínues, binaries, ...) i mantenir la coherència entre les imputacions mitjançant la implantació passiva.

S'aplica la funció *mice* a un subconjunt dels camps del dataset, no es contemplen els camps **PassengerId**, **Name**, **Ticket**, **Cabin**, **Survived** perquè no aporten res al model i es simplifiquen els càlculs

```
library('mice')
```

```
## Warning: package 'mice' was built under R version 3.4.4
```

```
library('randomForest')
```

```
## Warning: package 'randomForest' was built under R version 3.4.4
```

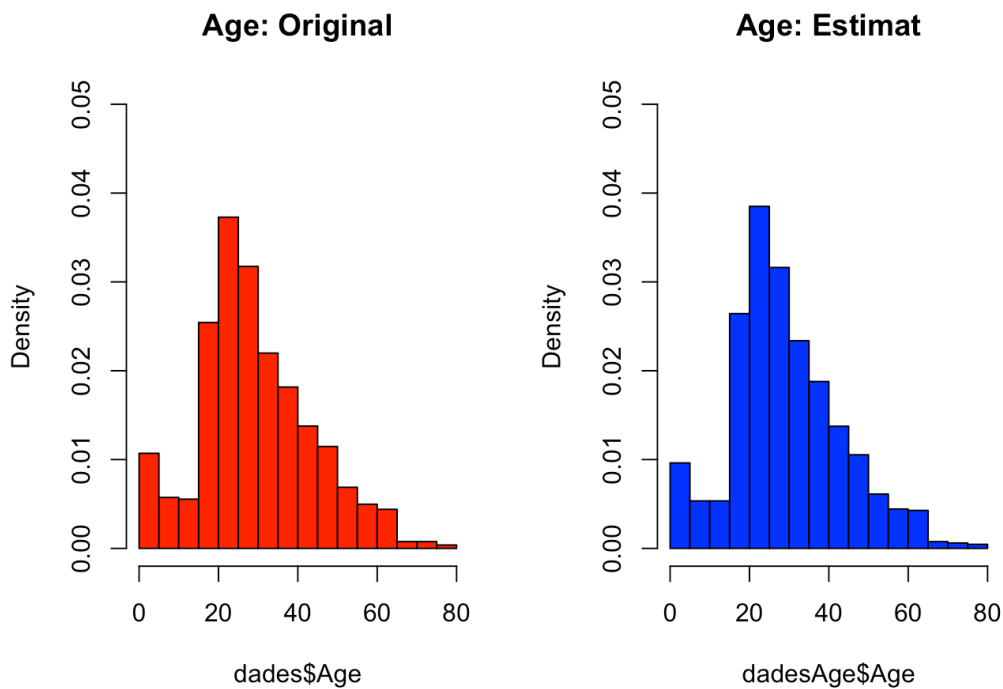
```
set.seed(1000)
daesAge <- complete(mice(daes[, !names(daes) %in% c('PassengerId', 'Name', 'Ticket', 'Cabin', 'Survived')], method='rf'))
```

```
##
## iter imp variable
## 1 1 Age
## 1 2 Age
## 1 3 Age
## 1 4 Age
## 1 5 Age
## 2 1 Age
## 2 2 Age
## 2 3 Age
## 2 4 Age
## 2 5 Age
## 3 1 Age
## 3 2 Age
## 3 3 Age
## 3 4 Age
## 3 5 Age
## 4 1 Age
## 4 2 Age
## 4 3 Age
## 4 4 Age
## 4 5 Age
## 5 1 Age
## 5 2 Age
## 5 3 Age
## 5 4 Age
## 5 5 Age
```

```
## Warning: Number of logged events: 25
```

Mitjançant la comparació visual de plots sobre l'**Age** original i el estimat, es pot verificar que la assignació de les edats no informades no introdueix biaixos en les dades.

```
par(mfrow=c(1,2))
hist(dades$Age, freq=F, main='Age: Original', col='red', ylim=c(0,0.05))
hist(dadesAge$Age, freq=F, main='Age: Estimat', col='blue', ylim=c(0,0.05))
```



```
dades$Age <- dadesAge$Age
summary (dades)
```

```
## PassengerId      Survived      Pclass      Name
## Min.   : 1      Min.   :0.0000      Min.   :1.000      Length:1309
## 1st Qu.: 328      1st Qu.:0.0000      1st Qu.:2.000      Class :character
## Median : 655      Median :0.0000      Median :3.000      Mode  :character
## Mean   : 655      Mean   :0.3838      Mean   :2.295
## 3rd Qu.: 982      3rd Qu.:1.0000      3rd Qu.:3.000
## Max.   :1309      Max.   :1.0000      Max.   :3.000
## NA's   :418
## Sex      Age      SibSp      Parch
## female:466 Min.   : 0.17      Min.   :0.0000      Min.   :0.000
## male :843  1st Qu.:21.00      1st Qu.:0.0000      1st Qu.:0.000
##          Median :28.00      Median :0.0000      Median :0.000
##          Mean   :29.74      Mean   :0.4989      Mean   :0.385
##          3rd Qu.:38.00      3rd Qu.:1.0000      3rd Qu.:0.000
##          Max.   :80.00      Max.   :8.0000      Max.   :9.000
##
## Ticket      Fare      Cabin      Embarked
## Length:1309      Min.   : 0.000      Length:1309      : 0
## Class :character 1st Qu.: 7.896      Class :character  C:272
## Mode  :character Median : 14.454      Mode  :character  Q:123
##          Mean   : 33.276                      S:914
##          3rd Qu.: 31.275
##          Max.   :512.329
##
```

- En el camp **Fare** s'observen diversos valors a 0 que es considera un valor fals, que correspon a valors no informats. Per estimar-los s'utilitza l'algoritme kNN. En primer lloc es substitueix els valors 0 per NA, ja que l'algoritme només estima els valors NA del dataset. A continuació s'aplica l'algoritme i s'obtenen valors pels valors no informats de **Fare**.

```
# Els valors 0 de Fare es converteixen en NA i es dedueixen amb l'algoritme KNN
dades$Fare[dades$Fare == 0] <- NA

# Es Comprova si hi ha NA,s a Fare
sapply(dades, function(x) sum(is.na(x)))
```

```
## PassengerId      Survived      Pclass      Name      Sex      Age
##          0          418          0          0          0          0
## SibSp      Parch      Ticket      Fare      Cabin      Embarked
##          0          0          0          17          0          0
```

S'observa que al **Fare** hi ha 17 registres no informats.

```
library(VIM)
```

```
## Warning: package 'VIM' was built under R version 3.4.4
```

```
# S'aplica l'algoritme kNN
dades.kNN <- kNN(dades[, !names(dades) %in% c('PassengerId', 'Name', 'Ticket', 'Cabin', 'Survived')])

# Es comprova si hi ha NA,s a Fare estimat
sapply(dades.kNN, function(x) sum(is.na(x)))
```

```
## Pclass      Sex      Age      SibSp      Parch
##          0          0          0          0          0
## Fare      Embarked      Pclass_imp      Sex_imp      Age_imp
##          0          0          0          0          0
## SibSp_imp      Parch_imp      Fare_imp      Embarked_imp
##          0          0          0          0
```

```
# Es comprova si hi ha zeros a Fare estimat
length(dades.kNN$Fare[dades.kNN$Fare == 0])
```

```
## [1] 0
```

```
# S'assignen els valors de Fare estimats amb kNN al dataset original
dades$Fare = dades.kNN$Fare
```

- Per millorar el dataset es planteja la creació de nous camps a partir de les dades disponibles :

En primer lloc es crea el camp **TamanyFamilia** format a partir de la suma dels camps **SibSp** i **Parch** (cònjuges, germans, pares i fills) més el propi passatger.

```
# Creació del camp TamanyFamilia
dades$TamanyFamilia <- dades$SibSp + dades$Parch + 1
```

A partir d'aquest camp es crea el camp **TipusFamilia**, de tipus factor, format a partir de rangs de nombre de membres de la família embarcats.

```
# Creació del camp TipusFamilia
dades$TipusFamilia[dades$TamanyFamilia == 1] <- 'solitari'
dades$TipusFamilia[dades$TamanyFamilia < 5 & dades$TamanyFamilia > 1] <- 'petita'
dades$TipusFamilia[dades$TamanyFamilia > 4] <- 'nombrosa'
# Conversió a factor
dades$TipusFamilia <- as.factor(dades$TipusFamilia)
```

També a partir del camp Age es crea el camp **TipusEdat**, de tipus factor, format a partir de rangs d'edats.

```
# Es crea el camp TipusEdat en funció de franges d'edat i es converteix en factor
dades$TipusEdat[dades$Age < 18] <- 'Menor'
dades$TipusEdat[dades$Age >= 18 & dades$Age <= 65] <- 'Adult'
dades$TipusEdat[dades$Age > 65] <- 'Ancià '
dades$TipusEdat <- as.factor(dades$TipusEdat)
```

En funció del tipus de metodologies i algoritmes a utilitzar es pot prioritzar l'ús d'atributs numèrics com en el cas d'**Age** o **TamanyFamilia**, o be, atributs de tipus factor com són **TipusEdat** o **TipusFamilia**, calculats a partir dels camps numèrics. També s'ha descartat l'ús del camp **Ticket** per que presenta una estructura bastant variable, no segueix un patró i correspon en gran part a codis de ticket autonumèrics que s'han considerats que aporten poca informació per a la qüestió plantejada. El camp **Cabin** també s'ha descartat perquè, si bé era un camp que potencialment podria aportar informació rellevant respecte a la supervivència en funció de les cabines, per exemple, degut a la seva ubicació, el nombre de valors no informats és molt elevat (més de 4/5 parts) el que fa poc recomanable el completar els valors no informats mitjançant les tècniques aplicades en altres casos. Finalment s'ha utilitzar el camp TamanyFamilia, enlloc dels camps SibSp i Parch, ja que n'és l'agrupació d'ambos, reduint la dimensionalitat del dataset. S'han mantinut els camps **PassengerId** i **Name** per claredat ja que no aporten valor a la identificació de la supervivència.

Per tant el dataset final, integrat i filtrat és el següent :

```
#dataset final integrat i filtrat
dades = dades[, !names(dades) %in% c('SibSp', 'Parch', 'Ticket', 'Cabin', 'TipusFamilia', 'TipusEdat')]
summary (dades)
```

```
## PassengerId      Survived      Pclass      Name
## Min.   :    1   Min.   :0.0000   Min.   :1.000   Length:1309
## 1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
## Median : 655   Median :0.0000   Median :3.000   Mode  :character
## Mean   : 655   Mean   :0.3838   Mean   :2.295
## 3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
## Max.   :1309   Max.   :1.0000   Max.   :3.000
##
##      NA's      :418
## Sex      Age      Fare      Embarked TamanyFamilia
## female:466 Min.   : 0.17   Min.   : 3.171   : 0   Min.   : 1.000
## male :843   1st Qu.:21.00   1st Qu.: 7.925   C:272   1st Qu.: 1.000
##           Median :28.00   Median : 14.500   Q:123   Median : 1.000
##           Mean   :29.74   Mean   : 33.644   S:914   Mean   : 1.884
##           3rd Qu.:38.00   3rd Qu.: 31.387           3rd Qu.: 2.000
##           Max.   :80.00   Max.   :512.329           Max.   :11.000
##
```

El diccionari de dades del dataset final, integrat i filtrat és el següent :

Variable	Descripció	Tipus	Valors
PassangerId	Identificador de passatger	Enter	
Survival	Identificador de supervivència	Factor	0 = No, 1 = Yes
Pclass	Classe del passatge	Factor	1 = 1a, 2 = 2ona, 3 = 3 ^a
Name	Nom del passatger	Text	
Sex	Sexe	Factor	1 = male, 2 = female
Age	Edat en anys	Nombre	
Fare	Tarifa del passatge	Nombre	
Embarked	Port d'Embarcament	Factor	C = Cherbourg Q = Queenstown S = Southampton
TamanyFamilia	Nombre de familiars embarcats	Enter	

3.2. Identificació i tractament de valors extrems.

Identifiquem els valors extrems de les variables **Age** i **Fare** a partir de la funció boxplot:

```
outliers_age <- boxplot.stats(dades$Age)$out
outliers_fare <- boxplot.stats(dades$Fare)$out
outliers_age
```

```
## [1] 66.0 65.0 65.0 71.0 70.5 70.0 65.0 64.0 65.0 71.0 64.0 80.0 70.0 70.0
## [15] 74.0 67.0 76.0 64.0 64.0 80.0 64.0
```

```
outliers_fare
```



```
## [1] 71.2833 263.0000 146.5208 82.1708 76.7292 80.0000 83.4750
## [8] 73.5000 263.0000 77.2875 247.5208 73.5000 77.2875 79.2000
## [15] 66.6000 69.5500 69.5500 146.5208 69.5500 113.2750 76.2917
## [22] 90.0000 83.4750 90.0000 79.2000 86.5000 512.3292 79.6500
## [29] 153.4625 135.6333 77.9583 78.8500 91.0792 151.5500 247.5208
## [36] 151.5500 110.8833 108.9000 83.1583 262.3750 164.8667 134.5000
## [43] 69.5500 135.6333 153.4625 133.6500 66.6000 134.5000 263.0000
## [50] 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
## [64] 91.0792 90.0000 78.2667 151.5500 86.5000 108.9000 93.5000
## [71] 221.7792 106.4250 71.0000 106.4250 110.8833 227.5250 79.6500
## [78] 110.8833 79.6500 79.2000 78.2667 153.4625 77.9583 69.3000
## [85] 76.7292 73.5000 113.2750 133.6500 73.5000 512.3292 76.7292
## [92] 211.3375 110.8833 227.5250 151.5500 227.5250 211.3375 512.3292
## [99] 78.8500 262.3750 71.0000 86.5000 120.0000 77.9583 211.3375
## [106] 79.2000 69.5500 120.0000 93.5000 90.0000 80.0000 83.1583
## [113] 69.5500 89.1042 164.8667 69.5500 83.1583 82.2667 262.3750
## [120] 76.2917 263.0000 262.3750 262.3750 263.0000 211.5000 211.5000
## [127] 221.7792 78.8500 221.7792 75.2417 151.5500 262.3750 83.1583
## [134] 221.7792 83.1583 83.1583 247.5208 69.5500 134.5000 227.5250
## [141] 73.5000 164.8667 211.5000 71.2833 75.2500 106.4250 134.5000
## [148] 136.7792 75.2417 136.7792 82.2667 81.8583 151.5500 93.5000
## [155] 135.6333 146.5208 211.3375 79.2000 69.5500 512.3292 73.5000
## [162] 69.5500 69.5500 134.5000 81.8583 262.3750 93.5000 79.2000
## [169] 164.8667 211.5000 90.0000 108.9000
```

Si observem els valors extrems identificats de les variables **Age** i **Fare**, són valors coherents i que no tenen perquè suposar un error en la mostra de dades. No obstant, si volguéssim eliminar els valors ho fariem mitjançant el codi següent:

```
#dades<-dades[-which(dades$Age %in% outliers_age),]
#dades<-dades[-which(dades$Fare %in% outliers_fare),]
```

4. Anàlisi de les dades.

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

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

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

En l'apartat anterior, hem integrat els conjunts d'entrenament i test, aquest fet ens ha estat útil per a obtenir una major mostra per omplir amb major confiança els buits o zeros del conjunt. No obstant això, en aquest apartat ens centrarem en la variable **Survived**. Aquesta variable ens indica si un individu va sobreviure o no a l'accident del titànic, només la tenim informada per a les dades del conjunt d'entrenament, per tant, només utilitzarem les dades d'aquest.

```
titanic_train <- dades[!is.na(dades$Survived),]
titanic_test <- dades[is.na(dades$Survived),]
```

Hipòtesi nul·la i alternativa

En aquest cas d'estudi, prenem per a hipòtesi nul·la que la mitjana de les variables numèriques **Age**, **TamanyFamilia**, **Fare**, **Pclass** és independent a la variable **Survived**. És a dir, les mitjanes de les variables anteriors pels passatgers sobrevivents i les que no són iguals.

$$H_0 : \mu_1 = \mu_0$$

Com a hipòtesi alternativa, es té que la mitjana de cadascuna de les variables dels passatgers supervivents i la dels no supervivents és diferent.

$$H_1 : \mu_1 \neq \mu_0$$

On μ_1 és la mitjana de cadascuna de les variables numèriques pels passatgers supervivents i μ_2 és la mitjana de la variable d'estudi pels passatgers no supervivents.

Assumpció de normalitat

Comprovar si es compleix l'assumpció de normalitat en les dades. Per a fer-ho, s'aplica el test Shapiro-Wilk. Si el p-valor del test Shapiro Wilk és superior a 0.05, implica que la distribució de les dades no és significativament diferent de la distribució normal. És a dir, podem assumir normalitat.

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$Age  
## W = 0.98099, p-value = 2.263e-09
```

```
shapiro.test(titanic_train$TamanyFamilia)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$TamanyFamilia  
## W = 0.61508, p-value < 2.2e-16
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$Fare  
## W = 0.51848, p-value < 2.2e-16
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$Pclass  
## W = 0.71833, p-value < 2.2e-16
```

```
length(titanic_train$Pclass)
```

```
## [1] 891
```

Com que el p-valor és inferior a 0,05 en tots els casos, es rebutja la hipòtesi nul·la del test de Shapiro-Wilk que confirma l'assumpció de normalitat en les dades.

No obstant això, com que es té que $N = 891$ com a conseqüència del teorema del límit central, es pot considerar que les dades segueixen una distribució normal.

Homogeneïtat de la variància

Les variàncies són desconegudes. A continuació, per a decidir si apliquem variàncies iguals o diferents, apliquem el test de Fligner-Killena.

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.4.4
```

```
## Warning: package 'carData' was built under R version 3.4.4
```

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

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: Age by Survived
## Fligner-Killeen:med chi-squared = 1.7343, df = 1, p-value = 0.1879
```

```
fligner.test(TamanyFamilia ~ Survived, data = titanic_train)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: TamanyFamilia by Survived
## Fligner-Killeen:med chi-squared = 19.647, df = 1, p-value =
## 9.317e-06
```

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

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: Fare by Survived
## Fligner-Killeen:med chi-squared = 93.468, df = 1, p-value <
## 2.2e-16
```

```
fligner.test(Pclass ~ Survived, data = titanic_train)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: Pclass by Survived
## Fligner-Killeen:med chi-squared = 23.648, df = 1, p-value =
## 1.157e-06
```

El p-valor és superior a 0,05 en la variable age i per tant en ella podem assumir la igualtat de variàncies. No obstant això, per la resta de variables el p-valor és inferior a 0,05 i per tant s'assumeix la no igualtat de variàncies.

Per tant, apliquem test t de dues mostres independents per a la diferència de mitjanes, variàncies desconegudes i iguals en el cas de l'edat i apliquem test de wilcoxon en el cas de la mida de la família, la tarifa i la classe.

```
t.test(Age ~ Survived, data = titanic_train, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: Age by Survived
## t = 1.5319, df = 889, p-value = 0.1259
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.420076 3.408488
## sample estimates:
## mean in group 0 mean in group 1
## 30.38798 28.89377
```

```
wilcox.test(TamanyFamilia ~ Survived,data = titanic_train, var.equal=TRUE)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: TamanyFamilia by Survived  
## W = 77659, p-value = 7.971e-07  
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Fare ~ Survived,data = titanic_train, var.equal=TRUE)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: Fare by Survived  
## W = 59732, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0
```

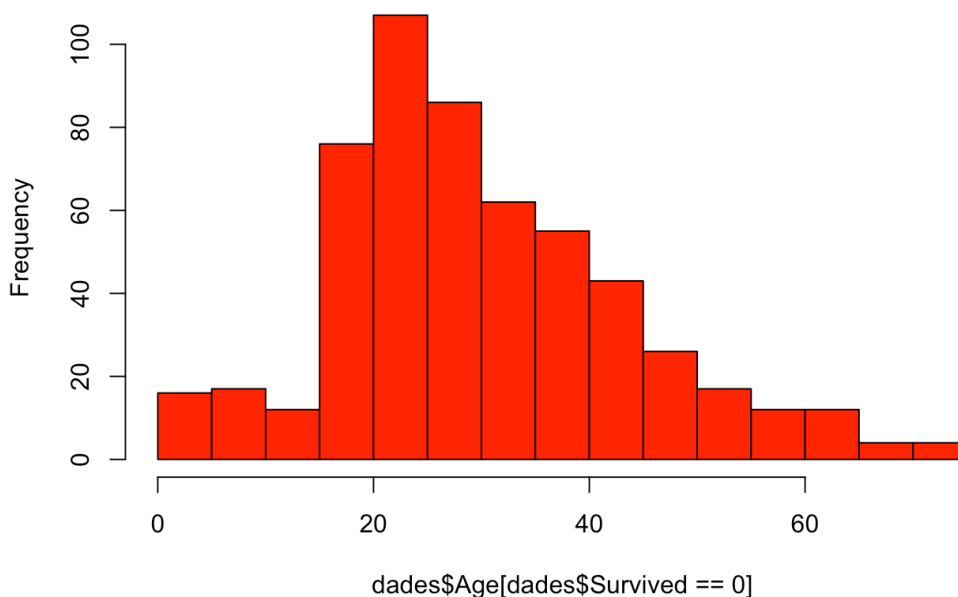
```
wilcox.test(Pclass ~ Survived,data = titanic_train, var.equal=TRUE)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: Pclass by Survived  
## W = 127940, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0
```

El valor-p és superior a 0,05 en el cas de l'edat i per tant s'accepta la hipòtesi H_0 i s'afirma que hi ha diferència estadística en la mitjana d'edat en els dos grups. A continuació es mostren dos histogrames, un amb l'edat dels supervivents i l'altre amb els no supervivents.

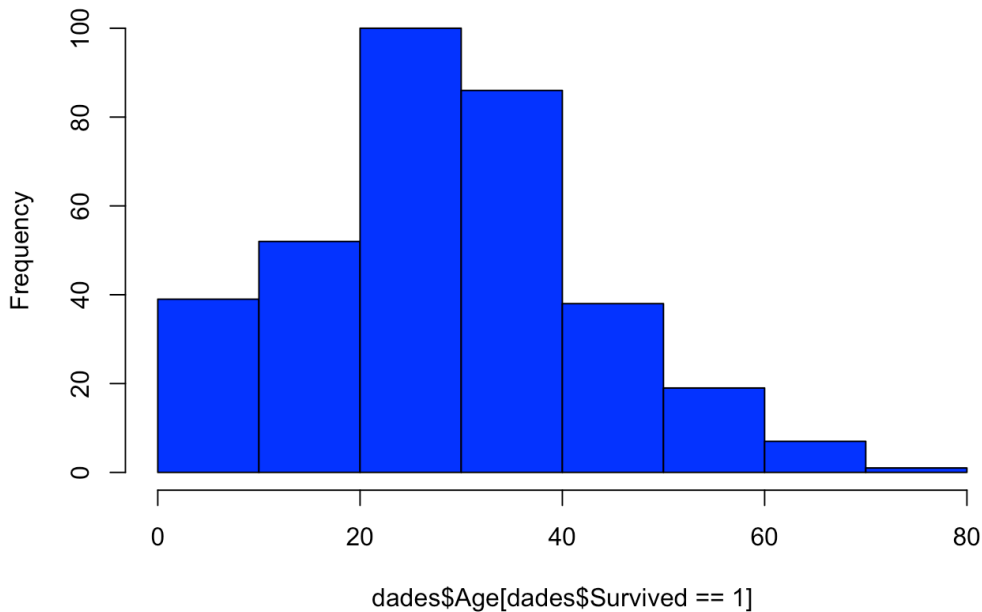
```
hist(dades$Age[dades$Survived == 0], main = "Histograma edat passatgers no supervivents", col = "red")
```

Histograma edat passatgers no supervivents



```
hist(dades$Age[dades$Survived == 1], main = "Histograma edat passatgers supervivents", col = "blue")
```

Histograma edat passatgers supervivents



En els dos histogrames anteriors, es

veu com la majoria de passatgers amb edat inferior a 20 anys van sobreviure al titànic. Pel que fa a les altres franges d'edat, no s'aprecien diferències.

En canvi, el valor-p és inferior a 0,05 en el cas del dimensió de la família, la tarifa i la classe, per tant es rebutja la hipòtesi H_0 i s'accepta la hipòtesi alternativa H_1 que afirmar que no hi ha diferència estadística de la mitjana de les dues variables en els dos grups.

Correlació

Per a comprovar la correlació entre variables, tan sols comparem les variables numèriques. Per tant, per a aquesta prova seleccionem tan sols les variables **Age**, **TamanyFamilia**, **Fare** i **Pclass**. En primer lloc calculem la matriu de correlació entre les variables.

```
titanic_cor <- data.frame(titanic_train$Age, titanic_train$TamanyFamilia, titanic_train$Fare, titanic_train$Pclass)
cor(titanic_cor)
```

```
##               titanic_train.Age titanic_train.TamanyFamilia
## titanic_train.Age              1.0000000             -0.3142511
## titanic_train.TamanyFamilia    -0.3142511             1.0000000
## titanic_train.Fare              0.0875904             0.2125432
## titanic_train.Pclass           -0.3543361             0.0659969
##               titanic_train.Fare titanic_train.Pclass
## titanic_train.Age              0.0875904            -0.3543361
## titanic_train.TamanyFamilia     0.2125432             0.0659969
## titanic_train.Fare              1.0000000            -0.5598987
## titanic_train.Pclass           -0.5598987             1.0000000
```

Estudiem doncs els casos amb major correlació:

-Variables **Pclass** i **Fare**: correlació de -0.5598987. -Variables **Pclass** i **Age**: correlació de -0.3543361. -Variables **Age** i **TamanyFamilia**: correlació de -0.3142511.

Abans de comprovar la correlació entre els parells de variables anteriors, per a decidir quin és el mètode més adequat a aplicar, cal comprovar si es compleix l'assumpció de normalitat en les dades. Per a fer-ho, tal com hem fet anteriorment, s'aplica el test Shapiro-Wilk.

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$Pclass  
## W = 0.71833, p-value < 2.2e-16
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$Age  
## W = 0.98099, p-value = 2.263e-09
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$Fare  
## W = 0.51848, p-value < 2.2e-16
```

```
shapiro.test(titanic_train$TamanyFamilia)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  titanic_train$TamanyFamilia  
## W = 0.61508, p-value < 2.2e-16
```

Com que en tots els casos, el p-valor és inferior a 0,05 es rebutja la hipòtesi nul·la del test de Shapiro-Wilk que confirma l'assumpció de normalitat en les dades. Per tant, per a estudiar la correlació entre els parells de variables estudiats aplicarem el test de Spearman.

```
cor.test(titanic_train$Pclass,titanic_train$Fare, method="spearman")
```

```
## Warning in cor.test.default(titanic_train$Pclass, titanic_train$Fare,  
## method = "spearman"): Cannot compute exact p-value with ties
```

```
##  
## Spearman's rank correlation rho  
##  
## data:  titanic_train$Pclass and titanic_train$Fare  
## S = 203770000, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
##      rho  
## -0.7284641
```

```
cor.test(titanic_train$Pclass,titanic_train$Age, method="spearman")
```

```
## Warning in cor.test.default(titanic_train$Pclass, titanic_train$Age, method  
## = "spearman"): Cannot compute exact p-value with ties
```

```
##
## Spearman's rank correlation rho
##
## data: titanic_train$Pclass and titanic_train$Age
## S = 158370000, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.3433576
```

```
cor.test(titanic_train$TmanyFamilia,titanic_train$Age, method="spearman")
```

```
## Warning in cor.test.default(titanic_train$TmanyFamilia, titanic_train
## $Age, : Cannot compute exact p-value with ties
```

```
##
## Spearman's rank correlation rho
##
## data: titanic_train$TmanyFamilia and titanic_train$Age
## S = 1.45e+08, p-value = 3.684e-12
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.2299892
```

En els tres casos, el p-valor és significatiu. Per tant, podem observar doncs una correlació prou significativa entre les parelles de variables estudiades, més concretament:

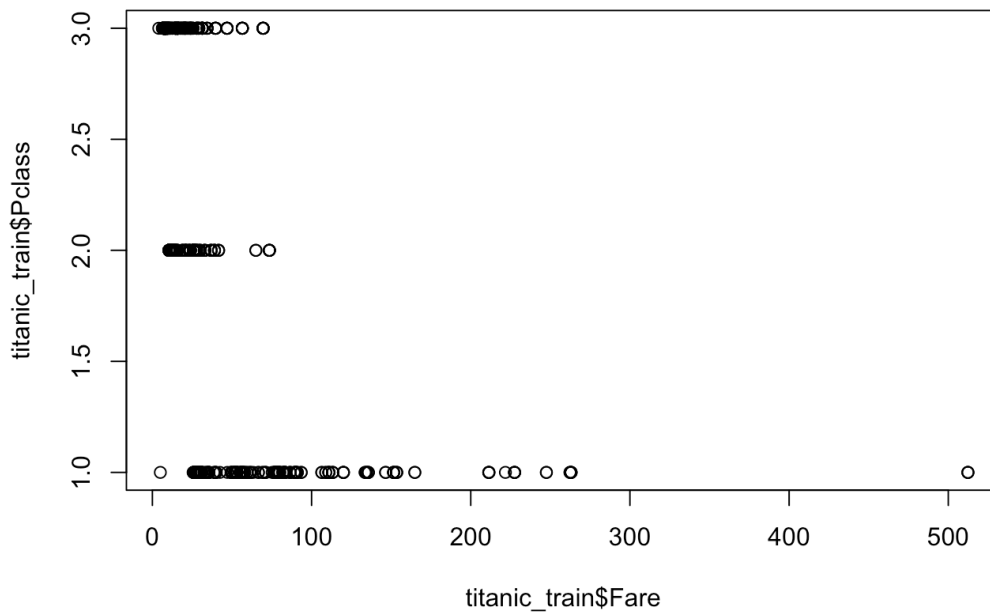
-Variables **Pclass** i **Fare**: correlació de -0.7284641 -Variables **Pclass** i **Age**: correlació de -0.3433576 -Variables **Age** i **TmanyFamilia**: correlació de -0.2299892

La primera resulta totalment comprensible ja que la tarifa sol anar marcada principalment per a la classe, on la primera és la que té tarifa més alta i la tercera la més baixa, és a dir correlació negativa.

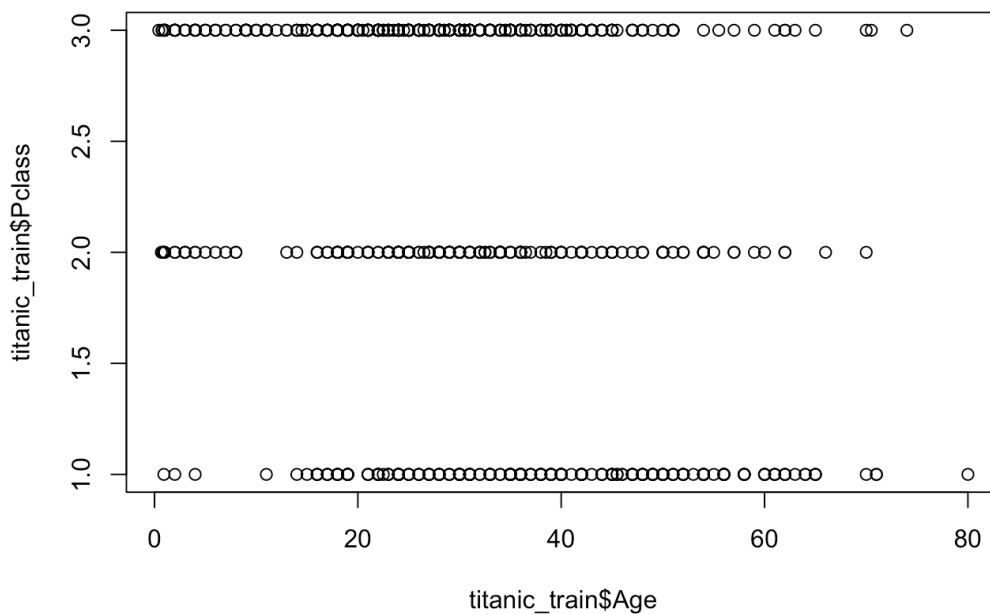
El segon resultat també sembla coherent, ja que com més edat té una persona, normalment acostuma a tenir major poder adquisitiu i més comoditat busca, per tant, selecciona classes superiors.

Les variables **Age** i **TmanyFamilia** tenen una petita correlació negativa que també té una possible explicació, ja que per exemple els nens, no viatjaran sols. En canvi, els adults sí que podrien viatjar sols. Finalment, observem els diagrames de dispersió de les 3 parelles anteriors.

```
plot(titanic_train$Fare, titanic_train$Pclass, main="Diagrama dispersió Pclass i Fare")
```

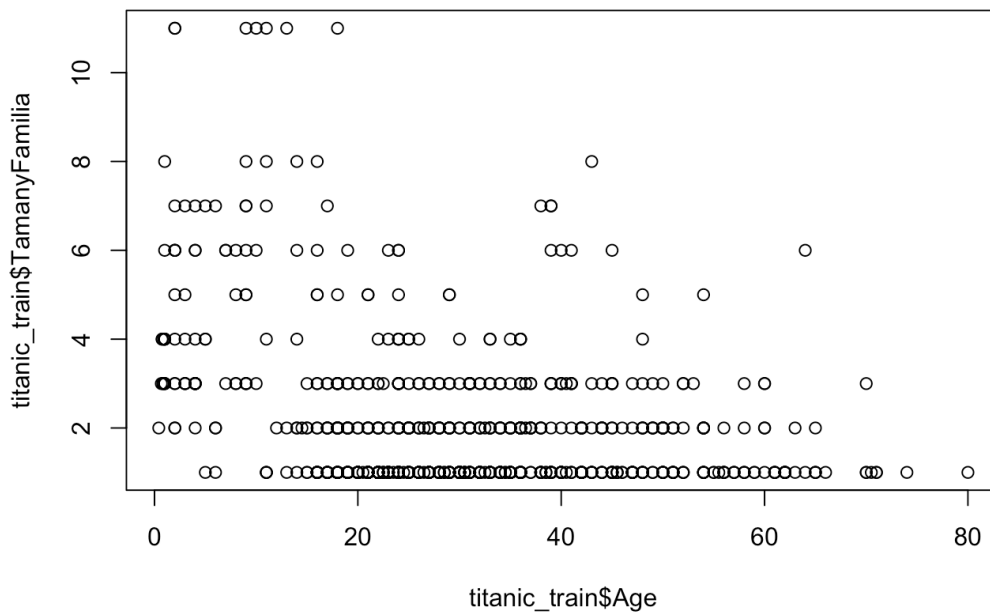
Diagrama dispersió Pclass i Fare

```
plot(titanic_train$Age, titanic_train$Pclass, main="Diagrama dispersió Pclass i Age")
```

Diagrama dispersió Pclass i Age

```
plot(titanic_train$Age, titanic_train$TamanyFamilia, main="Diagrama dispersió Age i TamanyFamilia")
```


Diagrama dispersió Age i TamanyFamilia



Regressió logística

En primer lloc cal tractar les variables categòriques. Usant el que s'anomena variables dummy podem calcular el model lineal. Per a fer-ho, especificarem quina és la categoria de referència amb la funció `relevel()`. En la variable **Sex**, la categoria de referència és "female" i per a la variable **Embarked** la categoria de referència és "S"

```
titanic_train$sexR <- relevel(titanic_train$Sex, ref = "female")
titanic_train$EmbarkedR <- relevel(titanic_train$Embarked, ref = "S")
```

Apliquem diversos models de regressió logística combinant diferents variables del conjunt de dades per a buscar el millor model de regressió. Per a buscar el millor model de regressió utilitzem la tècnica K-Cross-Validation. Aquesta tècnica permet dividir en diferents maneres els conjunts d'entrenament i test, comparant mitjanes o desviacions típiques per tal d'aconseguir un model més robust.

```
library('boot')
modell<- glm(Survived ~ sexR+TamanyFamilia+Age+EmbarkedR+Pclass,data=titanic_train)
cv.error1 = cv.glm(titanic_train,modell,K=10)$delta
summary(modell)
```

```
##
## Call:
## glm(formula = Survived ~ sexR + TamanyFamilia + Age + EmbarkedR +
##      Pclass, data = titanic_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00624  -0.21724  -0.07228   0.22000   0.98801
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.323701   0.066827  19.808 < 2e-16 ***
## sexRmale      -0.509715   0.027997 -18.206 < 2e-16 ***
## TamanyFamilia -0.031486   0.008508  -3.701 0.000228 ***
## Age          -0.005025   0.001023  -4.910 1.09e-06 ***
## EmbarkedRC     0.069263   0.033884   2.044 0.041235 *
## EmbarkedRQ     0.066257   0.047435   1.397 0.162831
## Pclass        -0.181467   0.017563 -10.332 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1446284)
##
##      Null deviance: 210.73  on 890  degrees of freedom
## Residual deviance: 127.85  on 884  degrees of freedom
## AIC: 814.69
##
## Number of Fisher Scoring iterations: 2
```

```
cv.error1
```

```
## [1] 0.1463620 0.1462094
```

```
model2<- glm(Survived ~ sexR+Age+Pclass,data=titanic_train)
cv.error2 = cv.glm(titanic_train,model2,K=10)$delta
summary(model2)
```

```
##
## Call:
## glm(formula = Survived ~ sexR + Age + Pclass, data = titanic_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05489  -0.23720  -0.07882   0.21561   0.97773
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2454421   0.0571957  21.775 < 2e-16 ***
## sexRmale      -0.4994160   0.0275270 -18.143 < 2e-16 ***
## Age          -0.0038999   0.0009843  -3.962 8.03e-05 ***
## Pclass        -0.1827548   0.0167452 -10.914 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1476099)
##
##      Null deviance: 210.73  on 890  degrees of freedom
## Residual deviance: 130.93  on 887  degrees of freedom
## AIC: 829.89
##
## Number of Fisher Scoring iterations: 2
```

```
cv.error2
```

```
## [1] 0.1480090 0.1479531
```

```
model3<- glm(Survived ~ sexR+EmbarkedR+Pclass,data=titanic_train)
cv.error3 = cv.glm(titanic_train,model3,K=10)$delta
summary(model3)
```

```
##
## Call:
## glm(formula = Survived ~ sexR + EmbarkedR + Pclass, data = titanic_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9774  -0.2303  -0.0772   0.2609   0.9228
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.04527    0.04345  24.056  <2e-16 ***
## sexRmale    -0.50886    0.02755 -18.469  <2e-16 ***
## EmbarkedRC   0.08515    0.03428   2.484   0.0132 *
## EmbarkedRQ   0.06059    0.04776   1.268   0.2050
## Pclass      -0.15307    0.01645  -9.307  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1491811)
##
##      Null deviance: 210.73  on 890  degrees of freedom
## Residual deviance: 132.17  on 886  degrees of freedom
## AIC: 840.32
##
## Number of Fisher Scoring iterations: 2
```

```
cv.error3
```

```
## [1] 0.1500152 0.1499269
```

```
model4<- glm(Survived ~ sexR+TamanyFamilia+Pclass,data=titanic_train)
cv.error4 = cv.glm(titanic_train,model4,K=10)$delta
summary(model4)
```

```
##
## Call:
## glm(formula = Survived ~ sexR + TamanyFamilia + Pclass, data = titanic_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9495  -0.2526  -0.1091   0.2047   1.0192
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.125021    0.043239  26.019 < 2e-16 ***
## sexRmale       -0.532079    0.027951 -19.036 < 2e-16 ***
## TamanyFamilia -0.021392    0.008226  -2.600  0.00946 **
## Pclass         -0.154148    0.015688  -9.826 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1490855)
##
##      Null deviance: 210.73  on 890  degrees of freedom
## Residual deviance: 132.24  on 887  degrees of freedom
## AIC: 838.76
##
## Number of Fisher Scoring iterations: 2
```

```
cv.error4
```

```
## [1] 0.1494668 0.1494115
```

```
model5<- glm(Survived ~ sexR+Age,data=titanic_train)
cv.error5 = cv.glm(titanic_train,model5,K=10)$delta
summary(model5)
```

```
##
## Call:
## glm(formula = Survived ~ sexR + Age, data = titanic_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7450  -0.1894  -0.1879   0.2580   0.8142
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.7391612    0.0356136  20.755 <2e-16 ***
## sexRmale       -0.5534355    0.0288230 -19.201 <2e-16 ***
## Age            0.0001031    0.0009723   0.106   0.916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1672435)
##
##      Null deviance: 210.73  on 890  degrees of freedom
## Residual deviance: 148.51  on 888  degrees of freedom
## AIC: 940.16
##
## Number of Fisher Scoring iterations: 2
```

```
cv.error5
```

```
## [1] 0.1677384 0.1676827
```

Dels models anteriors (i tots els provats) seleccionem el **model1**, ja que és el model que té major bondat per a tenir una menor AIC i també és el model amb un menor índex d'error entre els diferents conjunts de prova utilitzats en la validació creuada.

A partir d'aquest model predim la probabilitat de sobreviure de cadascun dels registres del conjunt de test:

```
titanic_test$sexR <- relevel(titanic_test$Sex, ref = "female")
titanic_test$EmbarkedR <- relevel(titanic_test$Embarked, ref = "S")

titanic_test$ProbSurvived <- predict(model1, titanic_test)
```

Mostrem la taula que conté el resultat de les prediccions del conjunt de test:

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.4.4
```

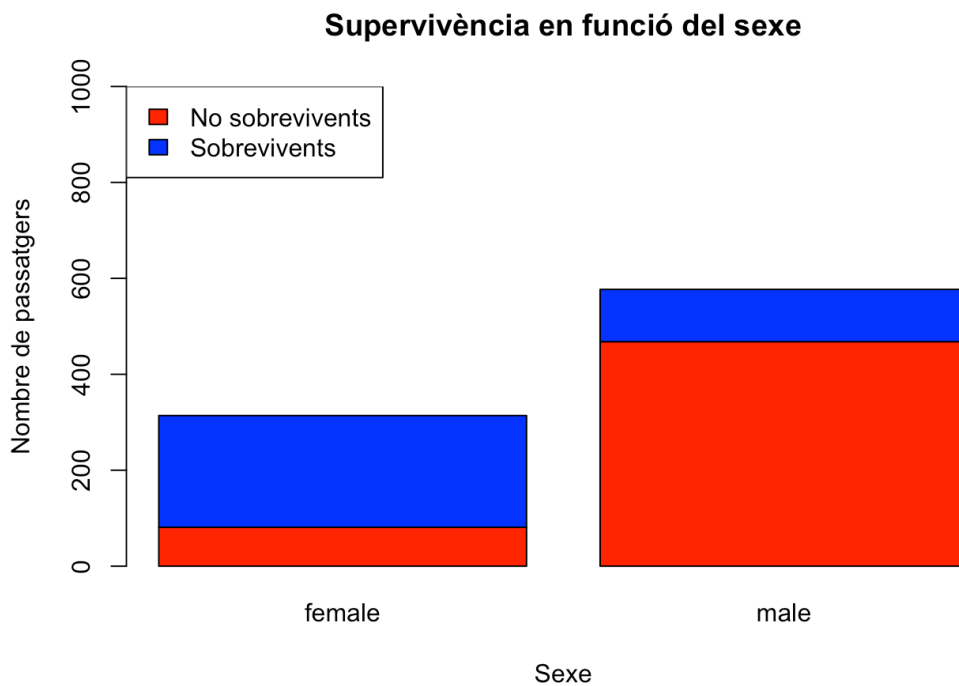
```
kable(head(titanic_test), format = 'markdown')
```

PassengerId	Survived	Pclass	Name	Sex	Age	Fare	Embarked	TamanyFamilia	sexR	EmbarkedR	ProbSurvived
892	892	NA	3 Kelly, Mr. James	male	34.5	7.8292	Q	1	male	Q	0.1310039
893	893	NA	3 Wilkes, Mrs. James (Ellen Needs)	female	47.0	7.0000	S	2	female	S	0.4801679
894	894	NA	2 Myles, Mr. Thomas Francis	male	62.0	9.6875	Q	1	male	Q	0.1742924
895	895	NA	3 Wirz, Mr. Albert	male	27.0	8.6625	S	1	male	S	0.1024324
896	896	NA	3 Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	12.2875	S	3	female	S	0.5742986
897	897	NA	3 Svensson, Mr. Johan Cervin	male	14.0	9.2250	S	1	male	S	0.1677532

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

Al llarg d'aquesta pràctica s'han mostrat diferents gràfiques o taules per a mostrar els resultats. Com que una de les conclusions de l'accident del titànic coneguda històricament és que les dones van obtenir un major índex de supervivència, afegim un gràfic que ens mostra la comparativa de la supervivència en funció del sexe.

```
counts = table(titanic_train$Survived, titanic_train$Sex)
barplot(counts, ylim=c(0, 1000), xlab="Sexe", ylab="Nombre de passatgers", main="Supervivència en funció del sexe",
col = c("red", "blue"))
legend("topleft", fill=c("red", "blue"), legend=c("No sobrevivents", "Sobrevivents"))
```



6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema?

6.1.- Procés de neteja i validació

Es partia de dos conjunts de dades relatives als passatgers del Titanic. Un conjunt de training (on s'informava el camp Survived) i un conjunt de test (on no s'informava el camp Survived). El dataset s'utilitza per generar models de predicció de la probabilitat de supervivència en funció de les característiques del passatger. Per a la fase de neteja i validació de les dades s'ha optat per integrar els dos fitxers per disposar de més registres que permetessin aplicar tècniques per completar camps no informats amb major precisió. D'una anàlisi preliminar s'ha comprovat que el conjunt de dades presentava nombrosos camps no informats (en especial el camp Age). S'ha procedit a analitzar aquests camps i aplicar diverses tècniques per completar-los.

El camp Fare (Tarifa) presentava un valor no informat. Per deduir-lo s'han obtingut tots els registres que compartien classe i origen d'embarcament, s'ha obtingut la mitja i s'ha assignat al valor no informat.

Per al camp Embarked (Origen d'embarcament), s'han obtingut els registres no informats i s'observa que tots ells comparteixen tarifa i classe. S'ha obtingut la mitjana de la tarifa pels passatgers de classe 1 per cada port d'origen i s'ha assignat el port origen que tenia una tarifa mitjana més propera.

En el cas del camp Age, donat que hi havia molts camps no informats i era important conservar-los, s'ha optat per aplicar la metodologia Mice (Multivariate Imputations by Chained Equations).

Finalment pel camp Fare, s'han substituït els valors 0 per valors no informats (per requeriment de l'algoritme kNN) i s'ha utilitzat l'algoritme kNN per estimar-los.

S'han creat camps nous com TamanyFamilia que és la agrupació dels camps SibSp i Parch, el que ha permès reduir la dimensió del dataset i s'han generat els camps factor TipusFamilia i TipusEdat per si eren interessants per l'aplicació de certs mètodes.

El dataset final està format pels camps PassengerId, Survival, Pclass, Name, Sex, Age, Fare, Embarked i TamanyFamilia.

6.2.- Anàlisi de dades

Fent l'anàlisi de dades hem arribat a la conclusió, mitjançant tests d'hipòtesis, que en general els passatgers del titànic menor de vint anys tenien un major índex de supervivència. Mitjançant aquesta mateixa prova hem vist que ni per la mida de la família, ni per la tarifa, ni per la classe en la qual es viatjava, s'aprecien diferències en l'índex de supervivència.

Aplicant un test de correlació, hem detectat correlació negativa entre el preu del bitllet i la classe que es viatja, l'edat del passatger i la classe en la qual viatge o bé l'edat del passatger i el nombre de familiars a bord.

Finalment i mitjançant un algoritme de regressió logística, hem escollit un model entre els diferents que hem creat que ens a permés predir la probabilitat de supervivència dels individus del conjunt de test.

Contribucions	Firma
Investigació prèvia	PC CR
Redacció de les respostes	PC CR
Desenvolupament codi	PC CR