# Práctica 2: Limpieza y análisis de datos

Pedro Uceda Martínez, Pablo Campillo Sánchez

3 de enero, 2021

## 1. Descripción del dataset

Durante esta práctica vamos a tratar el *dataset* base de la competición **Titanic - Machine Learning from Disaster**. En este conjunto de datos se nos presenta, para cada pasajero del tan famoso trasatlántico, sus datos personales más importantes, así como otros relacionados con su embarque en el Titanic, y si finalmente sobrevivieron al naufragio del mismo.

De este modo, este estudio es interesante dado que nos permite analizar cuáles fueron los factores que afectaron a la supervivencia de los pasajeros. Así, podremos, entre otras cosas, ver si solamente la clase del billete, el género y la edad condicionaron que un viajero se salvase tal y como hemos visto en la gran pantalla o bien hubiera habido otros factores que pudieran haber determinado la supervivencia del pasajero, como el número de billete.

Las variables de las que disponemos, para cada pasajero, son:

- **PassengerId**: Identificador artifical del pasajero.
- **Survived**: Si sobrevivió (1) o no (0).
- **Pclass**: Clase del pasaje.
- **Name**: Nombre del pasajero.
- **Sex**: Sexo del viajero.
- **Age**: Edad, en años.
- **SibSp**: Número de hermanos o esposas a bordo del Titanic.
- **Parch**: Número de padres / hijos a bordo del Titanic.
- **Ticket**: Número de ticket.
- **Fare**: Tarifa del pasaje.
- **Cabin**: Número de camarote.
- **Embarked**: Puerto desde el que embarcó el pasajero. Las posibles opciones son: Cherbourg (C), Queenstown (Q) o Southampton (S).

## 2. Integración y selección de los datos de interés a analizar.

Los datos a procesar provienen de **una única fuente**, por ello, **no es necesario realizar la fase de integración** o fusionado de los datos. En este apartado, primero **se cargarán los datos y se hará una exploración inicial** de los mismos para tener una idea más clara de como se distribuyen y, posteriormente, se procederá a **seleccionar los datos de interés** y a **generar nuevas características** que puedan resultar interesantes para el análisis posterior.

### 2.1 Exploración de los datos (screening)

A continuación, cargamos el **dataset**, sin **factors**, para evitar tratar los nombres de los pasajeros como tales.

```r
ds <- read.csv(file = "train.csv", header=TRUE, stringsAsFactors=FALSE)
str(ds)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Como se puede observar, el ***dataset*** contiene **891 registros y 12 atributos**. Están presentes las **variables cuantitativas PassengerId, Survived, Pclass, Age, SibSp, Parch y Fare**, todas tratadas como **int o num**. También tenemos las **variables cualitativas Ticket, PClass, Sex y Cabin, cargadas como cadena de caracteres**. **Survived**, aun siendo variable cuantitativa, representa 0 (No) y 1 (Yes), por lo que **en realidad es una variable cualitativa dicotómica**.

Para más claridad de los datos, procedemos a realizar las **siguientes transformaciones**:

- Transformamos el campo **Survived** a uno **categórico** con **dos valores, "Yes" y "Not"**, representando si el pasajero sobrevivió o no, respectivamente.
- Transformamos el campo cualitativo categórico **Embarked a un factor con 3 posibles valores**, cada uno con el nombre del puerto.
- Transformamos el **campo dicotómico Sex a un factor con 2 niveles, en lugar de trabajarlo como cadena de texto**.
- Transformamos el **campo Pclass**, que se ha cargado como campo cuantitativo, **a un factor con tres niveles, ordenado,** y le asignamos las etiquetas "1st", "2nd", "3rd".

```r
#Transformamos Survived a factor
ds$Survived <- factor(ds$Survived, levels=c(0, 1), labels = c("Not", "Yes"))

#Convertimos Embarked a factor con 3 niveles
ds$Embarked <- factor(ds$Embarked, levels=sort(c("C", "Q", "S")), labels = c("Cherbourg", "Queenstown",

#Convertimos Sex a factor con 2 niveles, female | male
ds$Sex <- factor(ds$Sex)

#Convertimos Pclass a un factor ordenado
ds$Pclass <- factor(ds$Pclass, ordered=TRUE, levels=c(1, 2, 3), labels=c("1st", "2nd", "3rd"))

#Revisamos como quedan los datos en el dataset
str(ds)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "Not","Yes": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Ord.factor w/ 3 levels "1st"<"2nd"<"3rd": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ 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/ 3 levels "Cherbourg","Queenstown",..: 3 1 3 3 3 2 3 3 3 1 ...
```

Para hacernos una idea de las características, vamos a mostrar las estadístcas básicas:

```r
summary(ds)
```

```
##   PassengerId    Survived  Pclass        Name               Sex
##  Min.   :  1.0   Not:549   1st:216   Length:891         female:314
##  1st Qu.:223.5   Yes:342   2nd:184   Class :character   male  :577
##  Median :446.0             3rd:491   Mode  :character
##  Mean   :446.0
##  3rd Qu.:668.5
##  Max.   :891.0
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare            Cabin                  Embarked
##  Min.   :  0.00   Length:891         Cherbourg  :168
##  1st Qu.:  7.91   Class :character   Queenstown : 77
##  Median : 14.45   Mode  :character   Southampton:644
##  Mean   : 32.20                      NA's       :  2
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

La información más relevante es:

- **Survived**: Hay más gente que falleció que sobrevivió.
- **Pclass**: Lo más común es tercera clases (Median).
- **Sex**: En el barco viajaban el doble de hombres que de mujeres.
- **age**: especifica la edad en años. Podemos ver que el mínimo es 0.42 años, así que se contemplan bebés. La persona más anciana tenía 80 años y la media de edad estaba en torno a los 30 años.
- **SibSp**: Lo más común es ir sin hermanos ni mujer.
- **Parch**: Es menos común todavía ir con descendientes o ascendientes.
- **Fare**: La media del precio del billete es 32.2 y la mediana 14. Esto indica que hay mucha disparidad de precios, siendo el máximo 512.
- **Embarked**: La mayoría embarcaron de Southamption, luego de Cherbourg y unos pocos de Queenstown.

Por último, hacemos una inspección visual de los campos que menos sabemos sobre ellos: Ticket y Cabin.

### 2.1.1 Campo Ticket

La codificación del billete (Ticket) parece que sigue diferentes patrones y además, hay viajeros que comparten el ticket ya que si los ordenamos, podemos comprobar que estos se repiten:

```r
sort(ds$Ticket)[1:10]
```

```
##  [1] "110152" "110152" "110152" "110413" "110413" "110413" "110465" "110465"
##  [9] "110564" "110813"
```

Si comprobamos los campos únicos, vemos que pasa de 891 a 681 valores diferentes.

```r
length(distinct(ds, Ticket)$Ticket)
```

```
## [1] 681
```

Además, el que un ticket se repita no depende de su tipo:

```r
aux <- count(ds, Ticket)
aux[order(aux[,2], decreasing = TRUE), ][1:10, ]
```

```
##          Ticket n
## 81         1601 7
## 334      347082 7
## 569    CA. 2343 7
## 250     3101295 6
## 338      347088 6
## 567     CA 2144 6
## 481      382652 5
## 622 S.O.C. 14879 5
## 34       113760 4
## 38       113781 4
```

Suponemos que se puede comprar un mismo billete para varias personas. ¿Compartirán el camarote? ¿Serán familia? Veamos los datos de estos 8.

Ticket 1601: Varias personas de origen chino tienen un billete común y, según los datos, no tienen parentesco entre sí.

```r
select(ds[ds$Ticket == "1601", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                Name Pclass    Fare Cabin    Embarked  Sex Age SibSp Parch
## 75    Bing, Mr. Lee    3rd 56.4958       Southampton male  32     0     0
## 170   Ling, Mr. Lee    3rd 56.4958       Southampton male  28     0     0
## 510  Lang, Mr. Fang    3rd 56.4958       Southampton male  26     0     0
## 644 Foo, Mr. Choong    3rd 56.4958       Southampton male  NA     0     0
## 693    Lam, Mr. Ali    3rd 56.4958       Southampton male  NA     0     0
## 827    Lam, Mr. Len    3rd 56.4958       Southampton male  NA     0     0
## 839 Chip, Mr. Chang    3rd 56.4958       Southampton male  32     0     0
```

**Ticket 347082:** Familia formada por 2 padres y 5 hijos de 2, 4, 6 y 9 años.

```r
select(ds[ds$Ticket == "347082", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                                          Name Pclass   Fare
## 14                                Andersson, Mr. Anders Johan    3rd 31.275
## 120                         Andersson, Miss. Ellis Anna Maria    3rd 31.275
## 542                      Andersson, Miss. Ingeborg Constanzia    3rd 31.275
## 543                         Andersson, Miss. Sigrid Elisabeth    3rd 31.275
## 611 Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren)    3rd 31.275
## 814                        Andersson, Miss. Ebba Iris Alfrida    3rd 31.275
## 851                   Andersson, Master. Sigvard Harald Elias    3rd 31.275
##     Cabin    Embarked    Sex Age SibSp Parch
## 14        Southampton   male  39     1     5
## 120       Southampton female   2     4     2
## 542       Southampton female   9     4     2
## 543       Southampton female  11     4     2
## 611       Southampton female  39     1     5
## 814       Southampton female   6     4     2
```

```
## 851       Southampton   male   4     4     2
```

**Ticket CA. 2343:** Deben ser hermanos viajando con sus esposas ya que tienen todas el mismo apellido y, aunque no se sabe la edad, el billete es caro (saldrían a 10 libras por cabeza)

```r
select(ds[ds$Ticket == "CA. 2343", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                   Name Pclass  Fare Cabin    Embarked    Sex Age
## 160        Sage, Master. Thomas Henry    3rd 69.55       Southampton   male  NA
## 181      Sage, Miss. Constance Gladys    3rd 69.55       Southampton female  NA
## 202              Sage, Mr. Frederick    3rd 69.55       Southampton   male  NA
## 325          Sage, Mr. George John Jr    3rd 69.55       Southampton   male  NA
## 793           Sage, Miss. Stella Anna    3rd 69.55       Southampton female  NA
## 847          Sage, Mr. Douglas Bullen    3rd 69.55       Southampton   male  NA
## 864 Sage, Miss. Dorothy Edith "Dolly"    3rd 69.55       Southampton female  NA
##     SibSp Parch
## 160     8     2
## 181     8     2
## 202     8     2
## 325     8     2
## 793     8     2
## 847     8     2
## 864     8     2
```

**Ticket 347088:** Matrimonio con sus 4 hijos de 2, 4, 9 y 10 años.

```r
select(ds[ds$Ticket == "347088", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                                Name Pclass Fare Cabin
## 64                              Skoog, Master. Harald    3rd 27.9
## 168 Skoog, Mrs. William (Anna Bernhardina Karlsson)    3rd 27.9
## 361                                 Skoog, Mr. Wilhelm    3rd 27.9
## 635                                 Skoog, Miss. Mabel    3rd 27.9
## 643                      Skoog, Miss. Margit Elizabeth    3rd 27.9
## 820                       Skoog, Master. Karl Thorsten    3rd 27.9
##        Embarked    Sex Age SibSp Parch
## 64  Southampton   male   4     3     2
## 168 Southampton female  45     1     4
## 361 Southampton   male  40     1     4
## 635 Southampton female   9     3     2
## 643 Southampton female   2     3     2
## 820 Southampton   male  10     3     2
```

**Ticket 3101295:** Madre con sus 5 hijos de 1, 2, 7, 14 y 16 años.

```r
select(ds[ds$Ticket == "3101295", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                      Name Pclass    Fare Cabin    Embarked
## 51              Panula, Master. Juha Niilo    3rd 39.6875       Southampton
## 165           Panula, Master. Eino Viljami    3rd 39.6875       Southampton
## 267              Panula, Mr. Ernesti Arvid    3rd 39.6875       Southampton
## 639 Panula, Mrs. Juha (Maria Emilia Ojala)    3rd 39.6875       Southampton
## 687               Panula, Mr. Jaako Arnold    3rd 39.6875       Southampton
## 825           Panula, Master. Urho Abraham    3rd 39.6875       Southampton
##        Sex Age SibSp Parch
## 51    male   7     4     1
## 165   male   1     4     1
## 267   male  16     4     1
```

```
## 639 female  41     0     5
## 687   male  14     4     1
## 825   male   2     4     1
```

**Ticket 347088:** Matrimonio con sus 4 hijos de 2, 4, 9 y 10 años.

```
select(ds[ds$Ticket == "347088", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                                Name Pclass Fare Cabin
## 64                             Skoog, Master. Harald    3rd 27.9
## 168 Skoog, Mrs. William (Anna Bernhardina Karlsson)    3rd 27.9
## 361                               Skoog, Mr. Wilhelm    3rd 27.9
## 635                               Skoog, Miss. Mabel    3rd 27.9
## 643                    Skoog, Miss. Margit Elizabeth    3rd 27.9
## 820                     Skoog, Master. Karl Thorsten    3rd 27.9
##        Embarked    Sex Age SibSp Parch
## 64  Southampton   male   4     3     2
## 168 Southampton female  45     1     4
## 361 Southampton   male  40     1     4
## 635 Southampton female   9     3     2
## 643 Southampton female   2     3     2
## 820 Southampton   male  10     3     2
```

**Ticket CA 2144:** Madre con sus 5 hijos de 1, 9, 11, 14 y 16 años.

```
select(ds[ds$Ticket == "CA 2144", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                        Name Pclass Fare Cabin    Embarked
## 60       Goodwin, Master. William Frederick    3rd 46.9       Southampton
## 72               Goodwin, Miss. Lillian Amy    3rd 46.9       Southampton
## 387         Goodwin, Master. Sidney Leonard    3rd 46.9       Southampton
## 481          Goodwin, Master. Harold Victor    3rd 46.9       Southampton
## 679 Goodwin, Mrs. Frederick (Augusta Tyler)    3rd 46.9       Southampton
## 684             Goodwin, Mr. Charles Edward    3rd 46.9       Southampton
##        Sex Age SibSp Parch
## 60    male  11     5     2
## 72  female  16     5     2
## 387   male   1     5     2
## 481   male   9     5     2
## 679 female  43     1     6
## 684   male  14     5     2
```

**Ticket 382652:** Madre con sus 4 hijos de 2, 4, 7 y 8 años.

```
select(ds[ds$Ticket == "382652", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                      Name Pclass   Fare Cabin   Embarked    Sex
## 17                   Rice, Master. Eugene    3rd 29.125       Queenstown   male
## 172                  Rice, Master. Arthur    3rd 29.125       Queenstown   male
## 279                    Rice, Master. Eric    3rd 29.125       Queenstown   male
## 788             Rice, Master. George Hugh    3rd 29.125       Queenstown   male
## 886 Rice, Mrs. William (Margaret Norton)    3rd 29.125       Queenstown female
##     Age SibSp Parch
## 17    2     4     1
## 172   4     4     1
## 279   7     4     1
## 788   8     4     1
## 886  39     0     5
```

**Ticket S.O.C. 14879:** Billete de 2a clase compartido entre hermanos e, imaginamos, que amigos.

```
select(ds[ds$Ticket == "S.O.C. 14879", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                            Name Pclass Fare Cabin    Embarked  Sex Age SibSp
## 73         Hood, Mr. Ambrose Jr    2nd 73.5       Southampton male  21     0
## 121 Hickman, Mr. Stanley George    2nd 73.5       Southampton male  21     2
## 386   Davies, Mr. Charles Henry    2nd 73.5       Southampton male  18     0
## 656   Hickman, Mr. Leonard Mark    2nd 73.5       Southampton male  24     2
## 666          Hickman, Mr. Lewis    2nd 73.5       Southampton male  32     2
##     Parch
## 73      0
## 121     0
## 386     0
## 656     0
## 666     0
```

La exploración del campo Ticket nos revela que los billetes se comparten, este hecho se ha confirmado tras estudiar un poco de historia en Wikipedia (https://en.wikipedia.org/wiki/Passengers_of_the_Titanic). Resulta que el precio típico de los billetes del Titanic era de 7, 13 y desde 30 libras tercera, segunda y primera clase, respectivamente. El precio de los niños de tercera era 3 libras. Por tanto, al menos, para hacer un análisis por persona, habría que:

- Adaptar el precio por persona: dividiendo fare por el número de personas que disponen del billete.
- Un campo nuevo podría indicar con cuántas personas se compartía el billete.
- Si del nombre nos quedamos con el apellido, podemos analizar también la probabilidad de muerte en función del apellido. ¿Hay apellidos más afortunados que otros o de clases sociales diferentes?

Por otro lado, al nombre del Ticket no hemos conseguido sacar una relación o significado claro a primera vista.

### 2.1.1 Campo Cabin

Al igual que con el campo Ticket, se han agrupado y contado los valores del campo Cabin. Como se puede ver en la tabla de abajo, la gran cantidad de registros no contiene el nombre del camarote (687). Los nombres de los camarotes parece que están formados por la letra de la cubierta (A-F) y seguido de un número. La mayoría de los registros corresponden a 1a clase, aunque también hay registros con 2a y 3a. También llama la atención, que recoge más de un camarote.

```
aux <- count(ds, Cabin)
head(aux[order(aux[,2], decreasing = TRUE), ])
```

```
##           Cabin   n
## 1               687
## 49      B96 B98   4
## 65  C23 C25 C27   4
## 147          G6   4
## 64      C22 C26   3
## 92            D   3
```

## 2.2 Selección y creación de características

Como hemos visto en el apartado anterior, tras explorar el campo Ticket, vimos que podíamos crear nuevos campos:

- **Surname**: campo del apellido del propietario del billete.
- **TicketOwners**: número de propietarios de un billete.
- **PricePerPerson**: precio del billete por persona, ya que Fare contiene el precio del billete total.

Los nombres están formados primero por el apellido, luego una coma y después el nombre. Para extraer el apellido, simplemente separamos por coma y nos quedamos con la primera parte:

```
library(tidyr)

ds <- separate(ds, Name, c("Surname", NA))
head(ds)
```

```
##   PassengerId Survived Pclass   Surname    Sex Age SibSp Parch           Ticket
## 1           1      Not    3rd    Braund   male  22     1     0        A/5 21171
## 2           2      Yes    1st   Cumings female  38     1     0         PC 17599
## 3           3      Yes    3rd Heikkinen female  26     0     0 STON/O2. 3101282
## 4           4      Yes    1st  Futrelle female  35     1     0           113803
## 5           5      Not    3rd     Allen   male  35     0     0           373450
## 6           6      Not    3rd     Moran   male  NA     0     0           330877
##      Fare Cabin    Embarked
## 1  7.2500       Southampton
## 2 71.2833   C85   Cherbourg
## 3  7.9250       Southampton
## 4 53.1000  C123 Southampton
## 5  8.0500       Southampton
## 6  8.4583        Queenstown
```

Luego, obtenemos el número de propietarios por billete (TicketOwners) y con este campo obtenemos el precio por persona (PricePerPerson). Los campos nuevos generados serían:

```
aux <- count(ds, Ticket)
ds <- merge(x = ds, y = aux, by = "Ticket", all.x = TRUE)
colnames(ds)[13] <- "TicketOwners"
ds$PricePerPerson <- ds$Fare / ds$TicketOwners
head(select(ds, Surname, TicketOwners, PricePerPerson))
```

```
##   Surname TicketOwners PricePerPerson
## 1  Cherry            3       28.83333
## 2  Rothes            3       28.83333
## 3  Maioni            3       28.83333
## 4 Taussig            3       26.55000
## 5 Taussig            3       26.55000
## 6 Taussig            3       26.55000
```

**Los atributos PassengerId, Ticket, Fare y Name no serán objeto de análisis**. Por tanto, los campos que finalmente se consideran para ser limipados y analizados son:

```
ds <- subset(ds, select = -c(PassengerId, Ticket, Fare) )
str(ds)
```

```
## 'data.frame':    891 obs. of  11 variables:
##  $ Survived      : Factor w/ 2 levels "Not","Yes": 2 2 2 2 1 2 1 1 2 2 ...
##  $ Pclass        : Ord.factor w/ 3 levels "1st"<"2nd"<"3rd": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Surname       : chr  "Cherry" "Rothes" "Maioni" "Taussig" ...
##  $ Sex           : Factor w/ 2 levels "female","male": 1 1 1 1 2 1 2 2 2 1 ...
##  $ Age           : num  30 33 16 39 52 18 47 NA 28 60 ...
##  $ SibSp         : int  0 0 0 1 1 0 0 0 0 1 ...
##  $ Parch         : int  0 0 0 1 1 2 0 0 0 0 ...
##  $ Cabin         : chr  "B77" "B77" "B79" "E67" ...
##  $ Embarked      : Factor w/ 3 levels "Cherbourg","Queenstown",..: 3 3 3 3 3 3 3 3 3 1 ...
##  $ TicketOwners  : int  3 3 3 3 3 3 2 2 1 1 ...
```

```
##  $ PricePerPerson: num  28.8 28.8 28.8 26.6 26.6 ...
```

# 3. Limpieza de datos

En este apartado vamos a limpiar los datos para que el análisis posterior y los modelos generados sean más representativos y correctos.

## 3.1 Elementos vacíos

Primero, vamos comprobar aquellos campos que son nulos o vacíos:

```r
#Estadísticas básicas
summary(ds)
```

```
##  Survived  Pclass      Surname               Sex           Age
##  Not:549   1st:216   Length:891         female:314   Min.   : 0.42
##  Yes:342   2nd:184   Class :character   male  :577   1st Qu.:20.12
##            3rd:491   Mode  :character                Median :28.00
##                                                      Mean   :29.70
##                                                      3rd Qu.:38.00
##                                                      Max.   :80.00
##                                                      NA's   :177
##      SibSp           Parch            Cabin                 Embarked
##  Min.   :0.000   Min.   :0.0000   Length:891         Cherbourg  :168
##  1st Qu.:0.000   1st Qu.:0.0000   Class :character   Queenstown : 77
##  Median :0.000   Median :0.0000   Mode  :character   Southampton:644
##  Mean   :0.523   Mean   :0.3816                      NA's       :  2
##  3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :8.000   Max.   :6.0000
##
##   TicketOwners   PricePerPerson
##  Min.   :1.000   Min.   :  0.000
##  1st Qu.:1.000   1st Qu.:  7.763
##  Median :1.000   Median :  8.850
##  Mean   :1.788   Mean   : 17.789
##  3rd Qu.:2.000   3rd Qu.: 24.288
##  Max.   :7.000   Max.   :221.779
##
```

```r
# Estadísticas de valores vacíos
colSums(is.na(ds))
```

```
##       Survived         Pclass        Surname            Sex            Age
##              0              0              0              0            177
##          SibSp          Parch          Cabin       Embarked   TicketOwners
##              0              0              0              2              0
## PricePerPerson
##              0
```

```r
colSums(ds=="")
```

```
##       Survived         Pclass        Surname            Sex            Age
##              0              0              0              0             NA
##          SibSp          Parch          Cabin       Embarked   TicketOwners
##              0              0            687             NA              0
## PricePerPerson
```

```
##                0
```

Vemos que los campos que tienen campos nulos o vacíos son:

- **Age** tiene 177 nulos y su valor debe ser mayor de cero. En este caso, lo ideal sería generar un modelo de regresión que predijese la edad ya que puede depender de la clase, el sexo pero sobre todo de la clase y el precio, ya que los niños pagan menos. Por simplicidad, vamos a hacer la mediana de las edades.
- **Embarked** tiene 2 valores nulos y cada persona tiene que haber embarcado desde algún puerto. En este caso, con el ticket a lo mejor se podría deducir desde donde se ha embarcado. En este caso, asignaremos el puerto más probable, es decir, desde donde más gente embarcó.
- **Cabin** tiene 687 valores vacíos y cada persona tiene que dormir en algún camarote. La cantidad de nulos es enorme, sobre todo para los de tercera clase. El camarote exacto no se puede averiguar. En base a la clase, se podría asignar una letra de cubierta. Pero para ello habría que cambiar la variable Cabin por Desk. En este caso, lo que haremos será eliminar la variable.

Como podemos comprobar, ya no hay nulos:

```r
age_median <- median(ds$Age, na.rm = TRUE)
ds[, 'Age'][is.na(ds[,'Age'])] <- age_median

embarked_most_frequent <-  levels(ds$Embarked)[which.max(ds$Embarked)]
ds[, 'Embarked'][is.na(ds[,'Embarked'])] <- embarked_most_frequent

ds <- subset(ds, select = -c(Cabin) )

summary(ds)
```

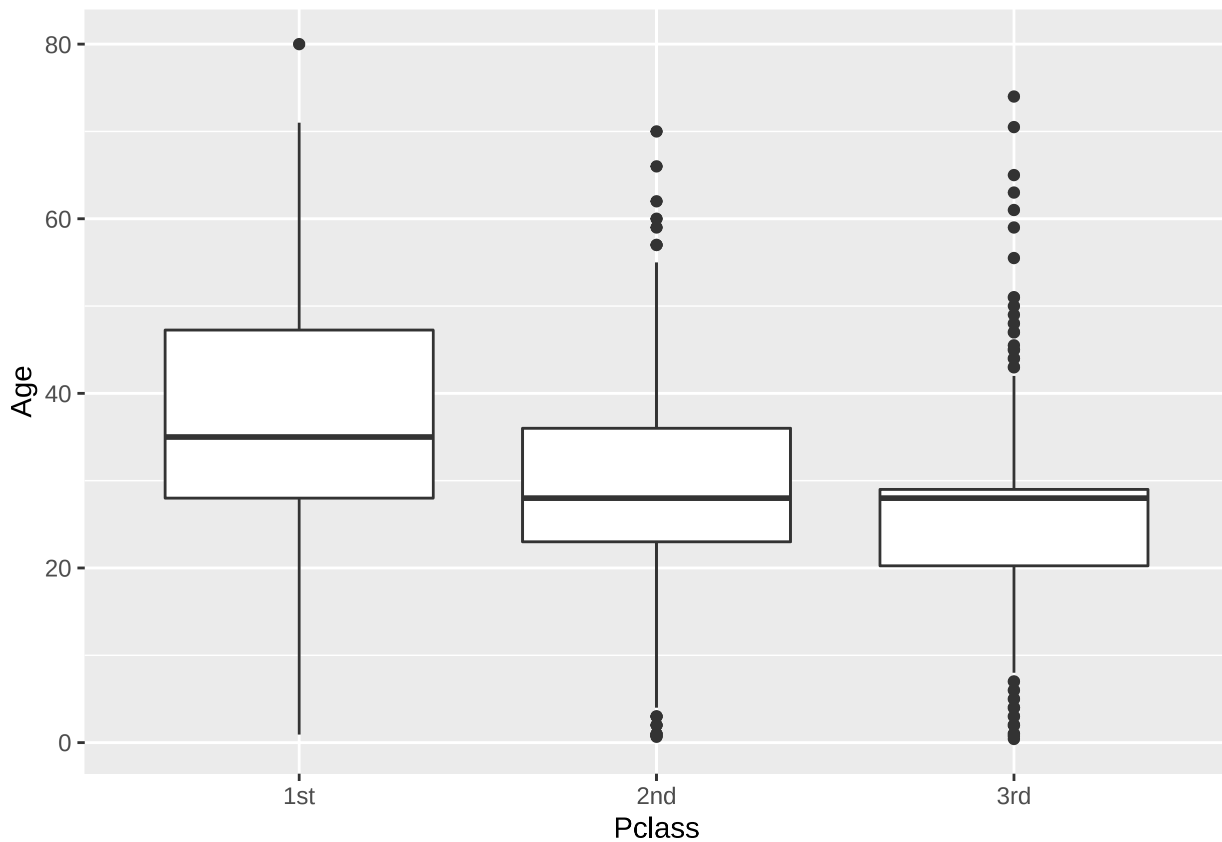
```
##  Survived  Pclass      Surname               Sex           Age
##  Not:549   1st:216   Length:891         female:314   Min.   : 0.42
##  Yes:342   2nd:184   Class :character   male  :577   1st Qu.:22.00
##            3rd:491   Mode  :character                Median :28.00
##                                                      Mean   :29.36
##                                                      3rd Qu.:35.00
##                                                      Max.   :80.00
##      SibSp           Parch                Embarked    TicketOwners
##  Min.   :0.000   Min.   :0.0000   Cherbourg  :170   Min.   :1.000
##  1st Qu.:0.000   1st Qu.:0.0000   Queenstown : 77   1st Qu.:1.000
##  Median :0.000   Median :0.0000   Southampton:644   Median :1.000
##  Mean   :0.523   Mean   :0.3816                     Mean   :1.788
##  3rd Qu.:1.000   3rd Qu.:0.0000                     3rd Qu.:2.000
##  Max.   :8.000   Max.   :6.0000                     Max.   :7.000
##  PricePerPerson
##  Min.   :  0.000
##  1st Qu.:  7.763
##  Median :  8.850
##  Mean   : 17.789
##  3rd Qu.: 24.288
##  Max.   :221.779
```

### 3.2 Identificación y tratamiento de valores extremos.

En este apartado vamos a analizar los valores de los campos numéricos para ver si hay valores que no tienen sentido o resultan extraños, por ejemplo, los valores extremos. Un criterio para identificar los valores extremos son aquellos que se sitúan a 3 veces la desviación estándar de la media. Una herramienta muy útil para identificar dichos valores son las gráficas de caja. Veamos por variables:

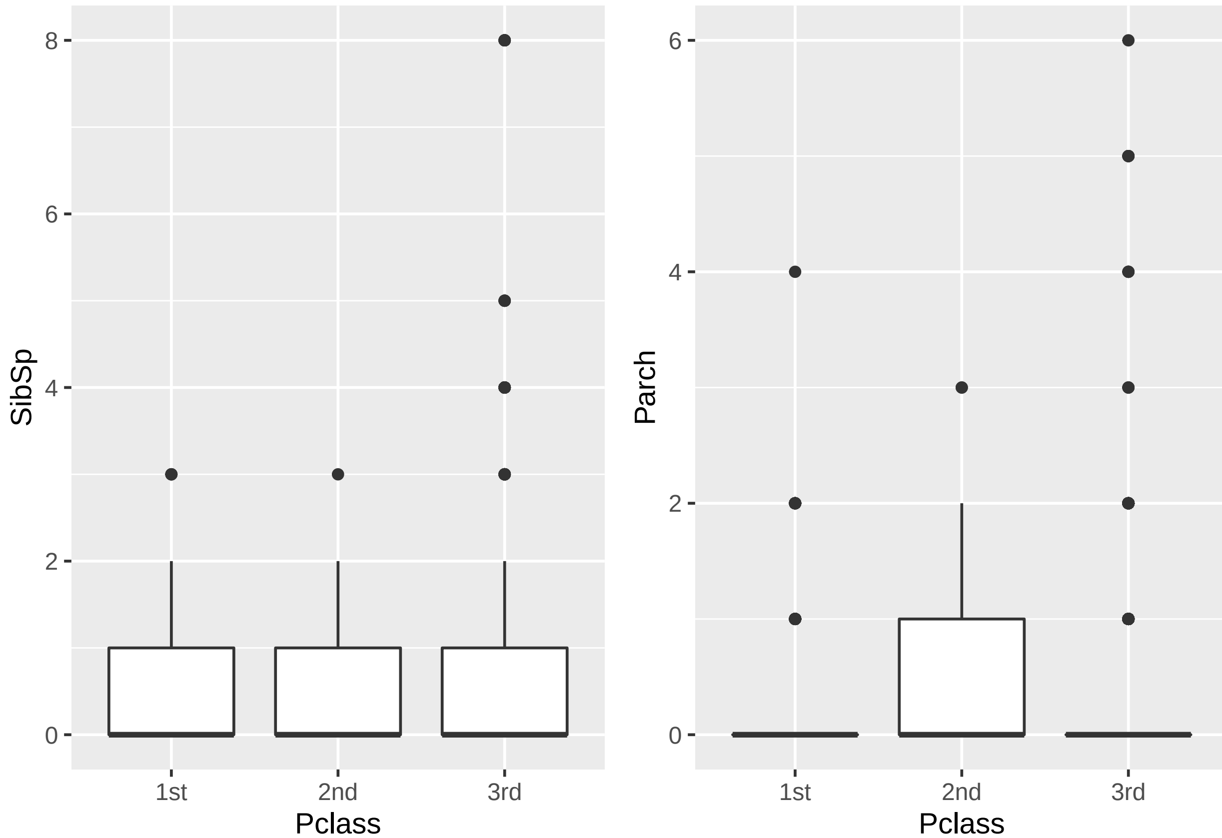
**Age**: Si hacemos las gráficas por clase, podemos ver que hay valores extremos pero están dentro de un rango de edades normal, entre 0.42 y 80. Se puede comprobar cómo, cuanto mejor es la clase, mayor es la edad.

```
gAge1 <- ggplot(ds, aes(x=Pclass, y=Age)) + geom_boxplot()
gAge1
```



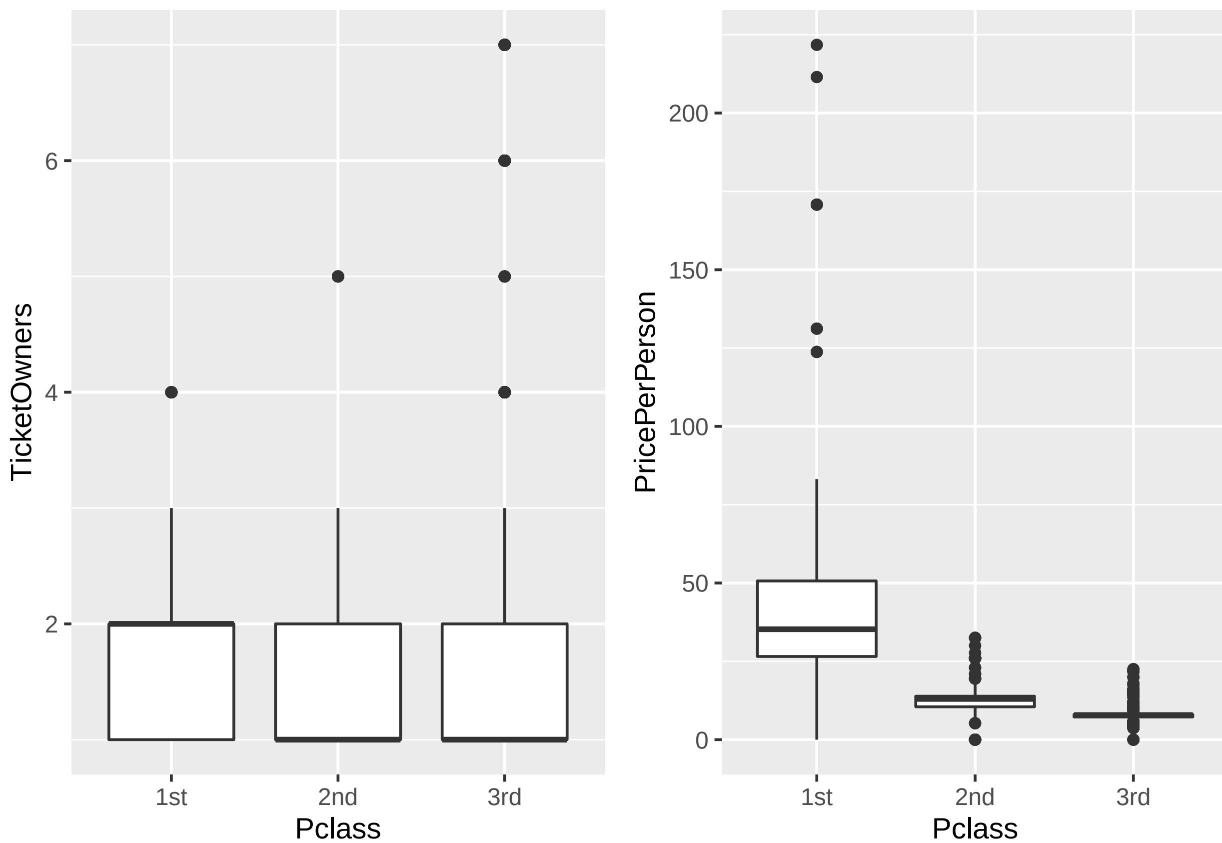

**SibSp** y **Parch**: En ambos casos se presentan outliers porque lo más común es viajar sin familiares y hay casos de 3, 4, 5 y hasta 8 hermanos que para la época no es extraño. Así que los valores para estos campos se consideran correctos.

```
gSibSp <- ggplot(ds, aes(x=Pclass, y=SibSp)) + geom_boxplot()
gParch <- ggplot(ds, aes(x=Pclass, y=Parch)) + geom_boxplot()
grid.arrange(gSibSp,gParch,nrow=1)
```




**TicketOwners** y **PricePerPerson**: Para la tercera y segunda clase, lo más común es viajar sólo y en primera es viajar con un acompañante. El número de personas máximo que comparten billete es 8 pero es normal si se considera que había alguna familiar con hasta 8 hermanos. Los datos son consistentes. Si vemos los precios por clase, vemos cómo son cada vez más elevados en función de la clase. Vemos outliers para tercera y segunda, por ejemplo billestes de tercera que cuestan 20 libras no debe estar bien. En cambio, para primera clase, no es raro que haya outliers, ya que el lujo nunca tiene techo. Según las investigaciones, hubo pasajeros de primera clase que llegaron a pagar más de 1000 libras.

```
gTicketOwners <- ggplot(ds, aes(x=Pclass, y=TicketOwners)) + geom_boxplot()
gPricePerPerson <- ggplot(ds, aes(x=Pclass, y=PricePerPerson)) + geom_boxplot()
grid.arrange(gTicketOwners,gPricePerPerson,nrow=1)
```




En este caso, lo que vamos a tratar, son aquellos casos en el que el precio del billete por persona es cero, ya que eso no puede ser a menos que la persona sea de la tripulación pero suponemos que son todo pasajeros. Por tanto, lo que haremos será reemplazar todos los valores 0 del campo PricePerPerson, por la mediana de dicho precio en función de a la clase que pertenezca el pasajero.

```
price_per_class <- aggregate(ds$PricePerPerson,         # Median by group
          list(ds$Pclass),
          median)
colnames(price_per_class) <- c("Pclass", "PricePerPerson")
price_per_class
```

```
##   Pclass PricePerPerson
## 1    1st        35.2500
## 2    2nd        13.0000
## 3    3rd         7.8542
```

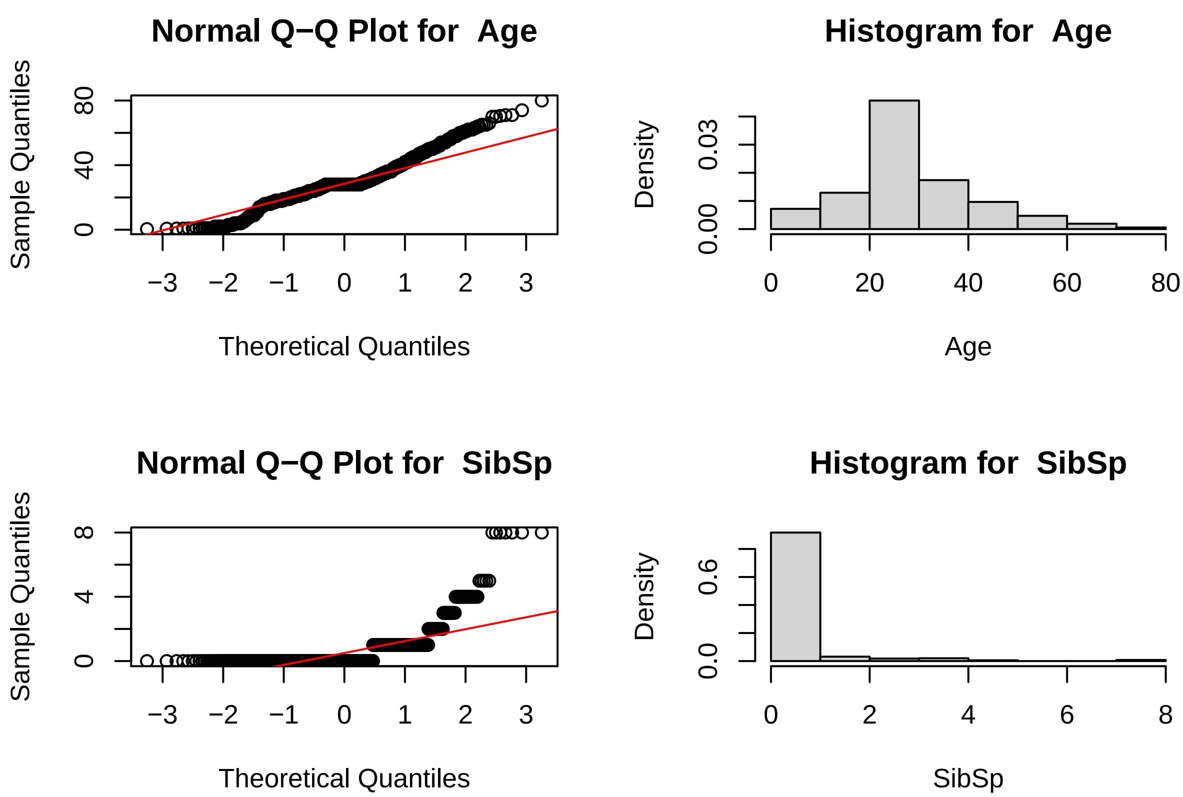
```
ds[ds$PricePerPerson == 0 & ds$Pclass == "1st", ]$PricePerPerson <- price_per_class[1, 2]
ds[ds$PricePerPerson == 0 & ds$Pclass == "2nd", ]$PricePerPerson <- price_per_class[2, 2]
ds[ds$PricePerPerson == 0 & ds$Pclass == "3rd", ]$PricePerPerson <- price_per_class[3, 2]
```

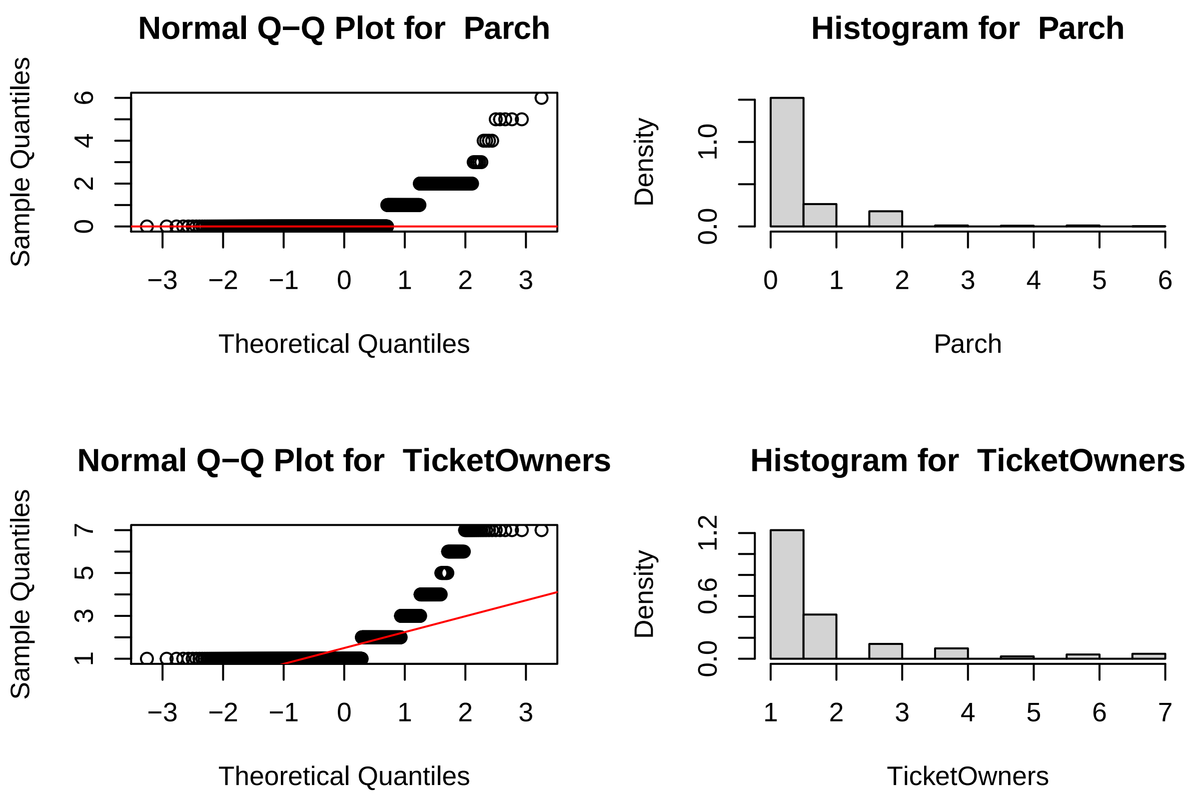
Con los datos limpios, procedemos a su análisis en el siguiente apartado.

# 4. Análisis de los datos

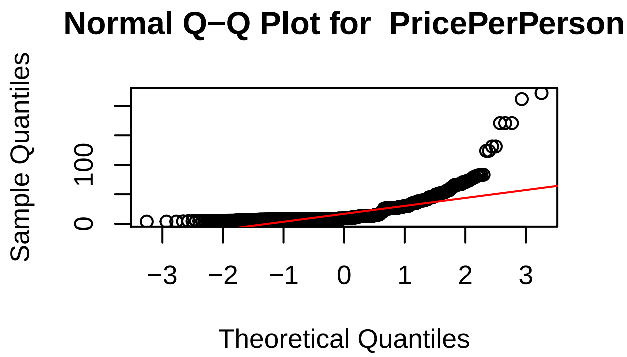
Antes de proceder a ver qué grupos de datos queremos normalizar, vamos a ver qué datos son normales y cuáles no, de manera gráfica…

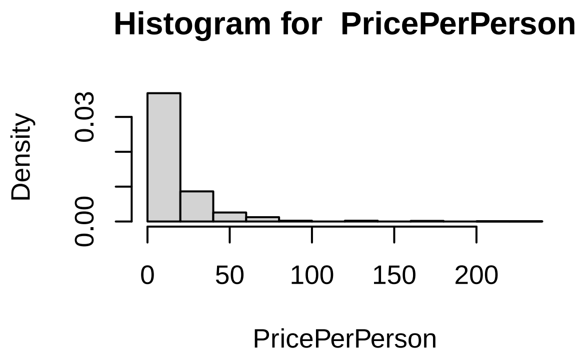
```
par(mfrow=c(2,2))
for(i in 1:ncol(ds)) {
  if (is.numeric(ds[,i])){
    qqnorm(ds[,i],main = paste("Normal Q-Q Plot for ",colnames(ds)[i]))
    qqline(ds[,i],col="red")
    hist(ds[,i],
      main=paste("Histogram for ", colnames(ds)[i]),
      xlab=colnames(ds)[i], freq = FALSE)
  }
}
```

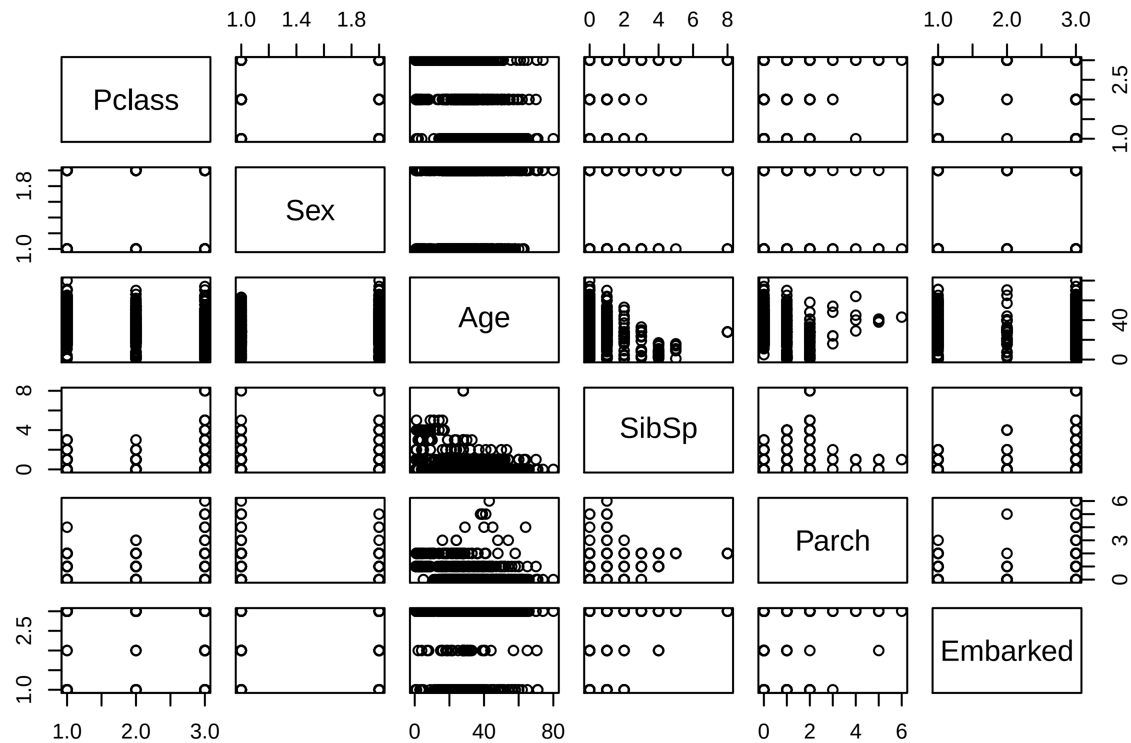





```r
plot(ds[,c("Pclass", "Sex", "Age", "SibSp", "Parch", "Embarked")])
```


Normal Q−Q Plot for PricePerPerson
Sample Quantiles
0
100
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Histogram for PricePerPerson
Density
0.00
0.03
0
50
100
150
200
PricePerPerson




### 4.1 Selección de los grupos de datos que se quieren analizar / comparar.

A continuación, se nombran los distintos grupos de datos que nos parecen interesantes:

- Analizaremos si **los niños**, entendiendo como tales los pasajeros que tenían 16 años o menos, **tuvieron la misma probabilidad de sobrevivir que los adultos o, por el contrario, más**. Compararemos los dos subgrupos de viajeros para responder a la siguientes hipótesis, teniendo **Ps(X) como la probabilidad de supervivencia del subgrupo X**:

$$H_0 : p_s(children) = p_s(adults)$$
$$H_1 : p_s(children) > p_s(adults)$$

- Intentaremos **aproximar los datos** utilizando **un modelo de regresión**. **Partiremos de la edad**, con la que habremos trabajado anteriormente, **y el sexo**, y veremos si podemos incluir una tercera variable que nos permita que mejore el comportamiento de nuestro modelo.
- «Nos faltan 1»

A continuación, creamos un dataset para los pasajeros que son niños y otro para los adultos. Utilizaremos tales dataset posteriormente para realizar el contraste de hipótesis.

```
children_passengers <- ds[ds$Age <= 16,]
adults_passengers <- ds[ds$Age > 16,]
```

### 4.2. Comprobación de la normalidad y homogeneidad de la varianza

Comprobamos si el atributo Age de los pasajeros, objeto de nuestro análisis, sigue una distribución normal, utilizando el test de Shapiro-Wilk:

```r
shapiro.test(ds$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ds$Age
## W = 0.9541, p-value = 4.651e-16
```

Obtenemos un **p-palor muy pequeño, menor al nivel de significancia 0.05**, por lo que podemos rechazar la hipótesis nula del test y asumimos que **la variable Age no sigue una distribución normal**.

Dado que la variable Age no sigue una distirbución normal, utilizaremos el **test de Fligner-Killeen** para comprobar la homocedasticidad de la variable:

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 5.706, df = 1, p-value = 0.01691
```

Observamos que dado el p-value obtenido, menor que 0.05, no podemos rechazar la hipótesis nula y concluimos que **la variable Age presenta una distribución homogénea de la varianza**.

Asimismo comprobamos si ambos subgrupos que vamos a comparar tienen la misma varianza:

```r
var.test(children_passengers$Age, adults_passengers$Age)
```

```
##
##  F test to compare two variances
##
## data:  children_passengers$Age and adults_passengers$Age
## F = 0.26025, num df = 99, denom df = 790, p-value = 6.71e-14
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1967717 0.3563239
## sample estimates:
## ratio of variances
##          0.2602506
```

Por el p-value obtenido, muy pequeño, y el ratio que nos devuelve el test concluimos que **la varianza no es la misma para los dos grupos de supervivientes** (niños y adultos).

## 4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos

### 4.3.1 Supervivencia de niños vs adultos

Aunque la variable Age presente una distribución de la varianza homogénea, no tiene una distribución normal, por lo que no podemos utilizar tests pamétricos para comparar ambos grupos de datos. Utilizaremos pues el **test de Wilcox, no paramétrico, para comprobar si los niños sobrevivieron más que los adultos**.

```r
  wilcox.test(children_passengers$Age, adults_passengers$Age, alternative = "greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
```

```
## data:  children_passengers$Age and adults_passengers$Age
## W = 0, p-value = 1
## alternative hypothesis: true location shift is greater than 0
```

Como vemos por el p-value con valor 1, el test nos arroja de manera decisiva que **los niños** (primer grupo) **sobrevivieron mucho más que los adultos** (segundo grupo).

A modo de comprobación, comprobamos que mediante la utilización del test obtenemos que para la hipotesis nula contraria:

```
wilcox.test(children_passengers$Age, adults_passengers$Age, alternative = "less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  children_passengers$Age and adults_passengers$Age
## W = 0, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
```

En este caso el test nos arroja un valor p muy pequeño, lo que nos permite rechazar la hipotesis nula, si la hiciesemos, de que los niños sobrevivieron significiamente menos que los adultos.

### 4.3.2 Modelo de regresión

Como hemos comentado en el apartado 4.1, comenzaremos a construir nuestro modelo de regresion con los atributos Age y Sex. Dado que la variable **Survived es una variable cualitativa categórica**, utilizamos un **modelo de regresión logística** en detrimento del lineal, ya que el rendimiento del primero es mejor en este caso.

Procedemos construir este primer modelo y ver cómo se comporta:

```
model.logist1 = glm(formula = Survived ~ Age + Sex, family=binomial(link=logit), data = ds)

summary(model.logist1)
```

```
##
## Call:
## glm(formula = Survived ~ Age + Sex, family = binomial(link = logit),
##     data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7019  -0.6532  -0.6373   0.7723   1.9304
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.189804   0.221918   5.361 8.26e-08 ***
## Age         -0.004738   0.006378  -0.743    0.458
## Sexmale     -2.505314   0.167450 -14.962  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  917.25  on 888  degrees of freedom
## AIC: 923.25
##
```

```
## Number of Fisher Scoring iterations: 4
```

Vemos por el estadístico de Wald que la variable **Sex (p-value<0.05) sí es estadísticamente significativa, pero Age (p-value>0.05) no**. Por lo tanto, procedemos a **quitar la variable Age del modelo**.

Del ***data screaning*** observamos que el **Pclass parecía tener relación con la supervivencia**, puesto que los pasajeros de primera y segunda clase sobrevivieron mucho más que los de tercera. Procedemos a **incluirlo en el modelo en detrimento del atributo Age** y vemos también el rendimiento del nuevo modelo:

```
model.logist2.formula = Survived ~ Sex + Pclass

model.logist2 = glm(formula = model.logist2.formula, family=binomial(link=logit), data = ds)

summary(model.logist2)
```

```
##
## Call:
## glm(formula = model.logist2.formula, family = binomial(link = logit),
##     data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1877  -0.7312  -0.4476   0.6465   2.1681
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.38264    0.14690   9.412   <2e-16 ***
## Sexmale     -2.64188    0.18410 -14.351   <2e-16 ***
## Pclass.L    -1.34739    0.15142  -8.898   <2e-16 ***
## Pclass.Q    -0.09373    0.16889  -0.555    0.579
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  826.89  on 887  degrees of freedom
## AIC: 834.89
##
## Number of Fisher Scoring iterations: 4
```

Podemos observar que la variable **Pclass es estadísticamente significativa** y vemos que **el modelo mejora, ya que el Akaike Information Criterion (AIC) es menor que en el primer modelo** que realizamos.

Probamos a incluir también la variable SibSp en el modelo, ya que de manera intutiva tiene sentido que los hombres que viajasen solos sobreviviesen más que los que viajasen con esposa.

```
model.logist3.formula = Survived ~ Sex + Pclass + SibSp

model.logist3 = glm(formula = model.logist3.formula, family=binomial(link=logit), data = ds)

summary(model.logist3)
```

```
##
## Call:
## glm(formula = model.logist3.formula, family = binomial(link = logit),
##     data = ds)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2572  -0.6733  -0.4713   0.6013   2.5182
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.56138    0.16444   9.495  < 2e-16 ***
## Sexmale     -2.74124    0.19048 -14.391  < 2e-16 ***
## Pclass.L    -1.31980    0.15194  -8.686  < 2e-16 ***
## Pclass.Q    -0.07035    0.17032  -0.413  0.67959
## SibSp       -0.24651    0.09468  -2.604  0.00922 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  819.15  on 886  degrees of freedom
## AIC: 829.15
##
## Number of Fisher Scoring iterations: 4
```

Vemos que **SibSp también es estadísticamente significativa y que mejora un poco el rendimiento del algoritmo**.

Probamos a incorporar del mismo modo la variable Parch:

```r
model.logist4 = glm(formula = Survived ~ Sex + Pclass + SibSp + Parch, family=binomial(link=logit), data

summary(model.logist4)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + SibSp + Parch, family = binomial(link = logit),
##     data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2677  -0.6835  -0.4727   0.5945   2.5325
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.58671    0.17428   9.105   <2e-16 ***
## Sexmale     -2.76028    0.19552 -14.117   <2e-16 ***
## Pclass.L    -1.32010    0.15205  -8.682   <2e-16 ***
## Pclass.Q    -0.06965    0.17038  -0.409   0.6827
## SibSp       -0.23255    0.09933  -2.341   0.0192 *
## Parch       -0.04985    0.11045  -0.451   0.6518
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  818.94  on 885  degrees of freedom
```

```
## AIC: 830.94
##
## Number of Fisher Scoring iterations: 4
```

Vemos que **la variable Parch no es estadísticamente significativa**, ya que su estadístico de Wald es mayor que 0.05, por lo que **la descartamos**. Comprobamos por último si el precio que pagó cada pasajero por el ticket mejoraría el modelo:

```r
model.logist5 = glm(formula = Survived ~ Sex + Pclass + SibSp + PricePerPerson, family=binomial(link=lo

summary(model.logist5)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + SibSp + PricePerPerson,
##     family = binomial(link = logit), data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2480  -0.6791  -0.4713   0.6026   2.5188
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.528908   0.201452   7.589 3.21e-14 ***
## Sexmale        -2.739235   0.190606 -14.371  < 2e-16 ***
## Pclass.L       -1.283190   0.200893  -6.387 1.69e-10 ***
## Pclass.Q       -0.084521   0.177780  -0.475  0.63449
## SibSp          -0.247118   0.094754  -2.608  0.00911 **
## PricePerPerson  0.001449   0.005209   0.278  0.78085
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  819.07  on 885  degrees of freedom
## AIC: 831.07
##
## Number of Fisher Scoring iterations: 4
```

Podemos observar que **la variable Fare tampoco es estadísticamente significativa, por lo que también la eliminamos del modelo**.

Tras este proceso, podemos concluir que **el mejor modelo logístico que explica la variable Survived es nuestro tercer modelo**, que utiliza Age , Pclass y SibSp para explicar la variable Survived:

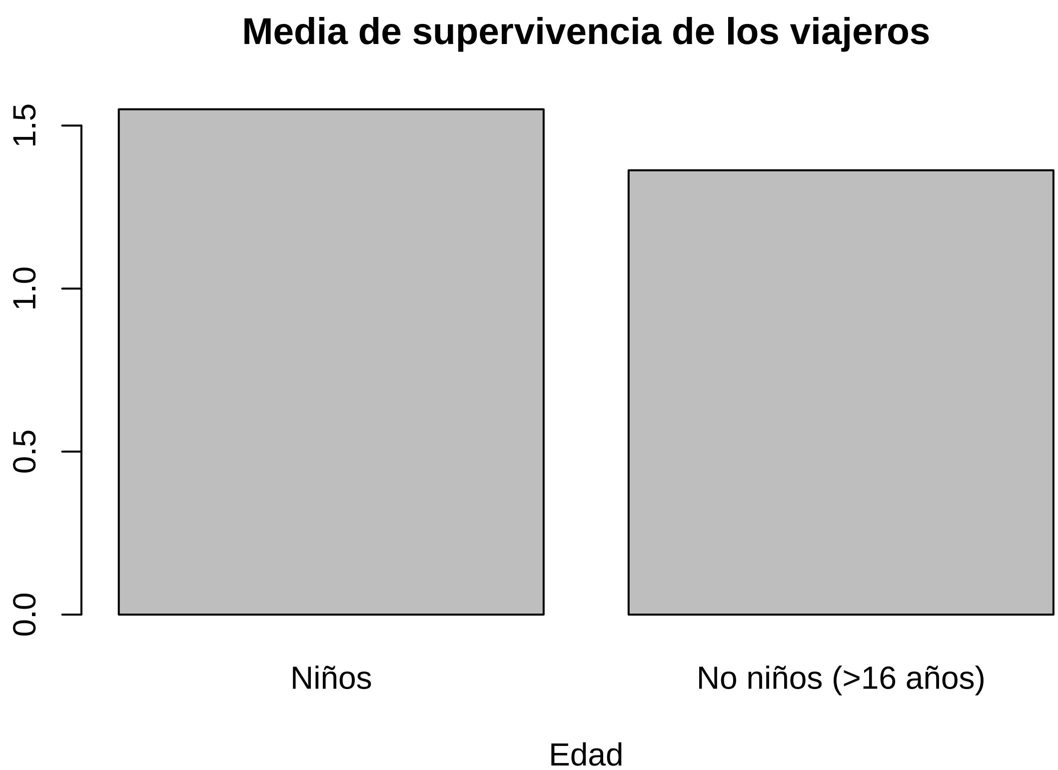
$$Survived = exp(3.43 - 2.74 * Sexmale - 0.93 * Pclass - 0.24 * SibSp)$$

# 5. Representación de los resultados a partir de tablas y gráficas

## 5.1 Comparación entre menores de 16 años y mayores de 16 años

En el apartado anterior, hemos visto que **los niños sobrevivieron mucho más que los adultos**. Podemoso **visualizar** esto de manera gráfica:

```r
#Calculamos la media para los dos tipos de pasajeros y lo pintamos en un diagrama de barras

children_passengers$Survived <-as.integer(children_passengers$Survived)
adults_passengers$Survived <- as.integer(adults_passengers$Survived)

mean_children_passengers <- mean(children_passengers$Survived)
mean_adults_passengers <- mean(adults_passengers$Survived)

#Print it
barplot(c(mean_children_passengers, mean_adults_passengers), names =c("Niños", "No niños (>16 años)"), m
```



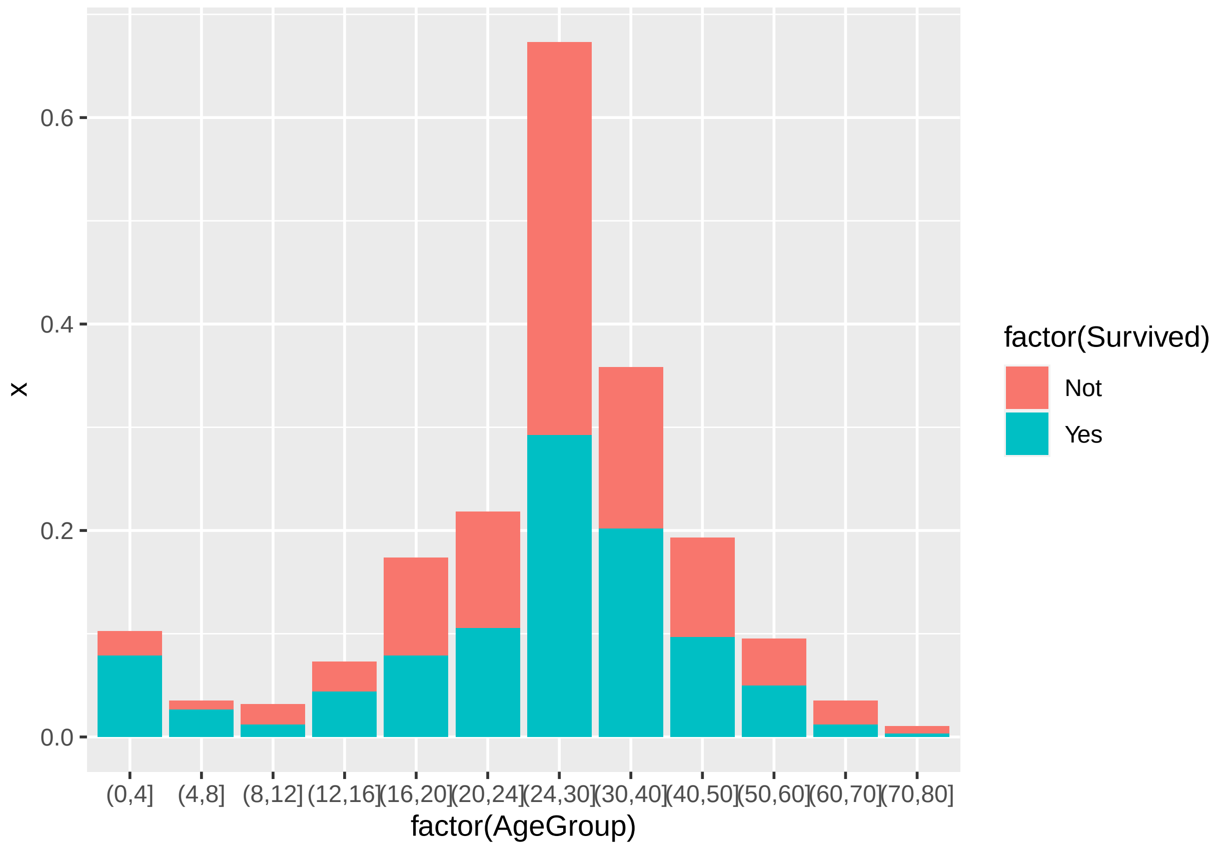

Podemos ver también **cómo se distribuye la supervivencia, agrupando los pasajeros por edades**:

```r
#Agrupamos por tramos de edad
ds$AgeGroup <- cut(ds$Age, breaks=c(0,4,8,12,16,20,24,30,40,50,60,70,80))

#Pintamos AgeGroup and Survived
sumAgeGroup <- summarize( group_by(ds, AgeGroup), n=length(AgeGroup))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
gAgeGroup1 <- ds %>%
group_by(Survived, AgeGroup) %>%
tally() %>%
group_by(Survived) %>%
mutate(x = n / sum(n)) %>%
ggplot() +
```

```r
geom_col(aes(
x = factor(AgeGroup),
y = x,
fill = factor(Survived)
), position = "stack")

grid.arrange(gAgeGroup1, nrow=1)
```




Puede verse que **para los pasajeros con 16 años o menos la supervivencia es significativamente mayor, con la excepción del rango de edad de 4 a 8 años**. Por lo tanto, la supervivencia de los niños es mayor, pero tiene **más dispersión que la de los adultos**.

### 5.2 Modelo de regresión logística

Vemos los coeficiented del modelo que hemos dado como mejor (el tercero) para ver cómo se comportan las variables que lo explican:

```r
exp(coefficients(model.logist3))
```

```
## (Intercept)     Sexmale    Pclass.L    Pclass.Q       SibSp
##  4.76539580  0.06449016  0.26718830  0.93206951  0.78152046
```

```r
##IC
exp(confint(model.logist3))
```

```
## Waiting for profiling to be done...

##                  2.5 %     97.5 %
```

```
## (Intercept) 3.48107697 6.63840587
## Sexmale     0.04401861 0.09296145
## Pclass.L    0.19744638 0.35845643
## Pclass.Q    0.66820012 1.30385355
## SibSp       0.64213086 0.93250417
```

La variable Sex tiene un OR de 0.064, la Pclass un OR de 0.39 y la SibSp un 0.78, por lo que a la hora de explicar la variable Survived sorprendentemente tiene mucho más peso la variable SibSp que el sexo o la clase, si bien tiene un Intervalo de Confianza, con una confianza del 95%, más amplio que las otras dos variables.

Procedemos a ver cómo se comportaría nuestro modelo de regresión logística a clase y SibSp constante y distinto sexo:

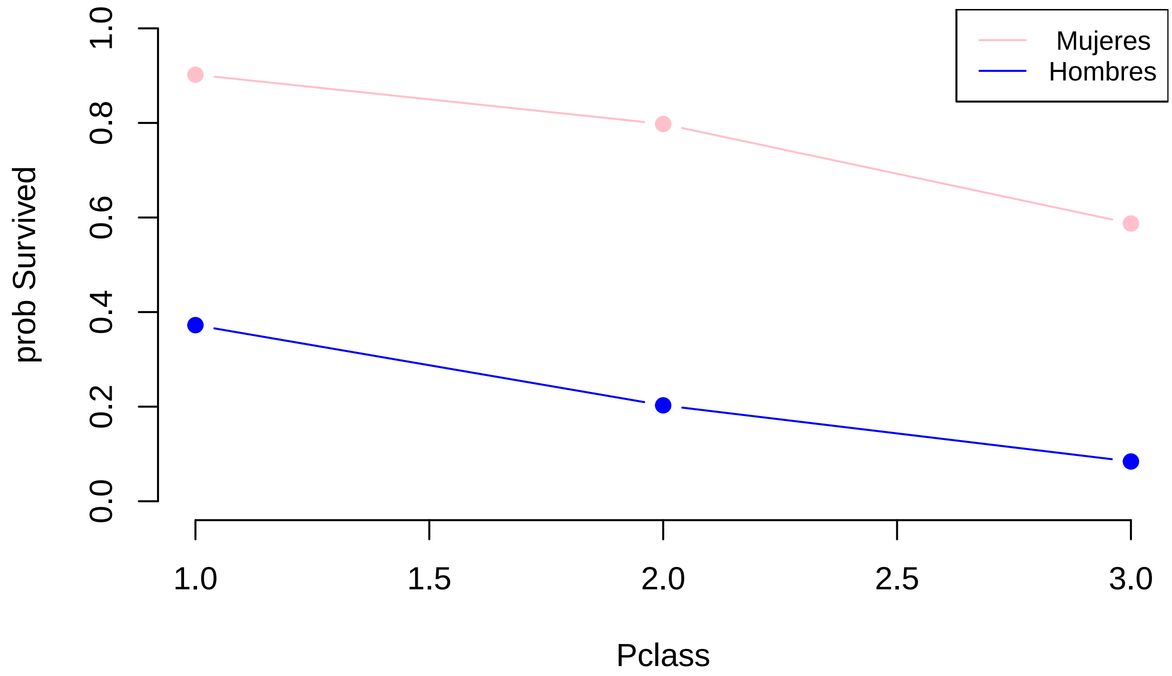
```
#Males
new_passengers_male <- data.frame(
  Sex = rep("male", times = 3),
  Pclass = c("1st","2nd","3rd"),
  SibSp = c(1,1,1)
)

#Females
new_passengers_female <- data.frame(
  Sex = rep("female", times = 3),
  Pclass = c("1st","2nd","3rd"),
  SibSp = c(1,1,1)
)

prob_males <- predict(model.logist3, newdata = new_passengers_male, type="response")

prob_females <- predict(model.logist3, newdata = new_passengers_female, type="response")

prob_males
```

```
##          1          2          3
## 0.37241865 0.20279162 0.08406647
```

```
prob_females
```

```
##         1         2         3
## 0.9019771 0.7977524 0.5873222
```

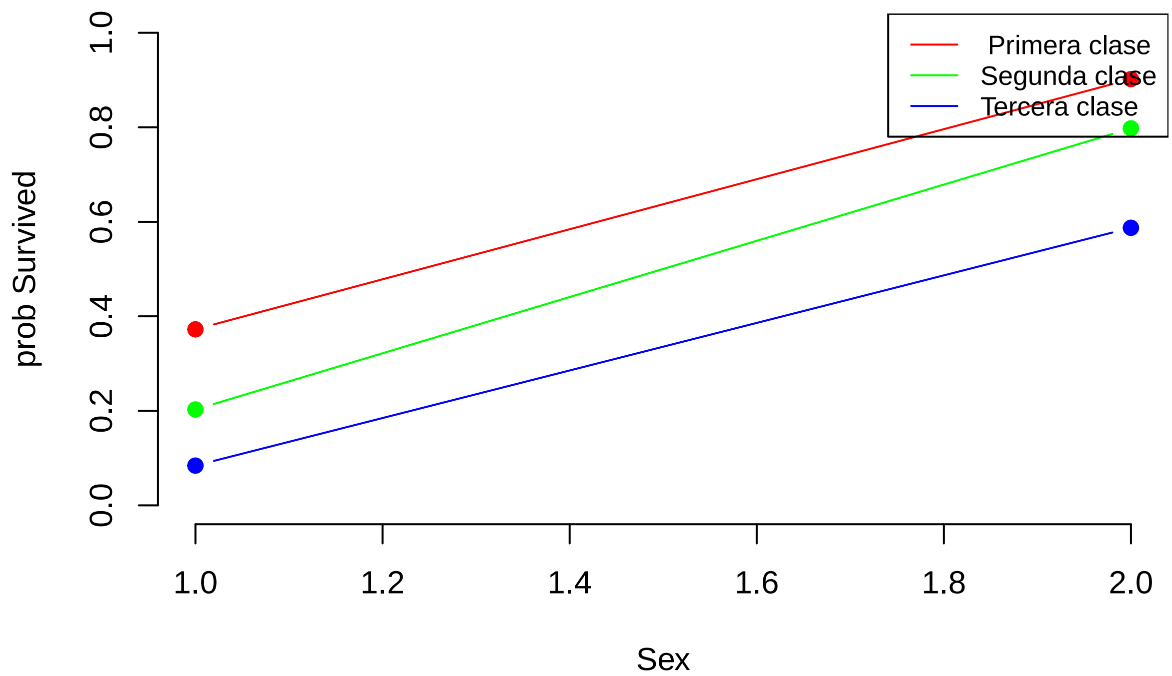
```
plot(c(1,2,3), prob_females, type = "b", frame = FALSE, pch = 19, col = "pink", xlab = "Pclass", ylab =

lines(c(1,2,3), prob_males, pch = 19, col = "blue", type = "b")

legend("topright", legend=c(" Mujeres", "Hombres"), col=c("pink", "blue"), lty = c(1,1), cex=0.8)
```




Ahora con clase y SibSps constantes y distinto sexo:

```
new_passengers_class_1 <- data.frame(
  Sex = c("male", "female"),
  Pclass = c("1st","1st"),
  SibSp = c(1,1)
)

new_passengers_class_2 <- data.frame(
  Sex = c("male", "female"),
  Pclass = c("2nd","2nd"),
  SibSp = c(1,1)
)

new_passengers_class_3 <- data.frame(
  Sex = c("male", "female"),
  Pclass = c("3rd","3rd"),
  SibSp = c(1,1)
)

prob_1 <- predict(model.logist3, newdata = new_passengers_class_1, type="response")

prob_2 <- predict(model.logist3, newdata = new_passengers_class_2, type="response")

prob_3 <- predict(model.logist3, newdata = new_passengers_class_3, type="response")

plot(c(1, 2), prob_1, type = "b", frame = FALSE, pch = 19, col = "red", xlab = "Sex", ylab = "prob Survi
```

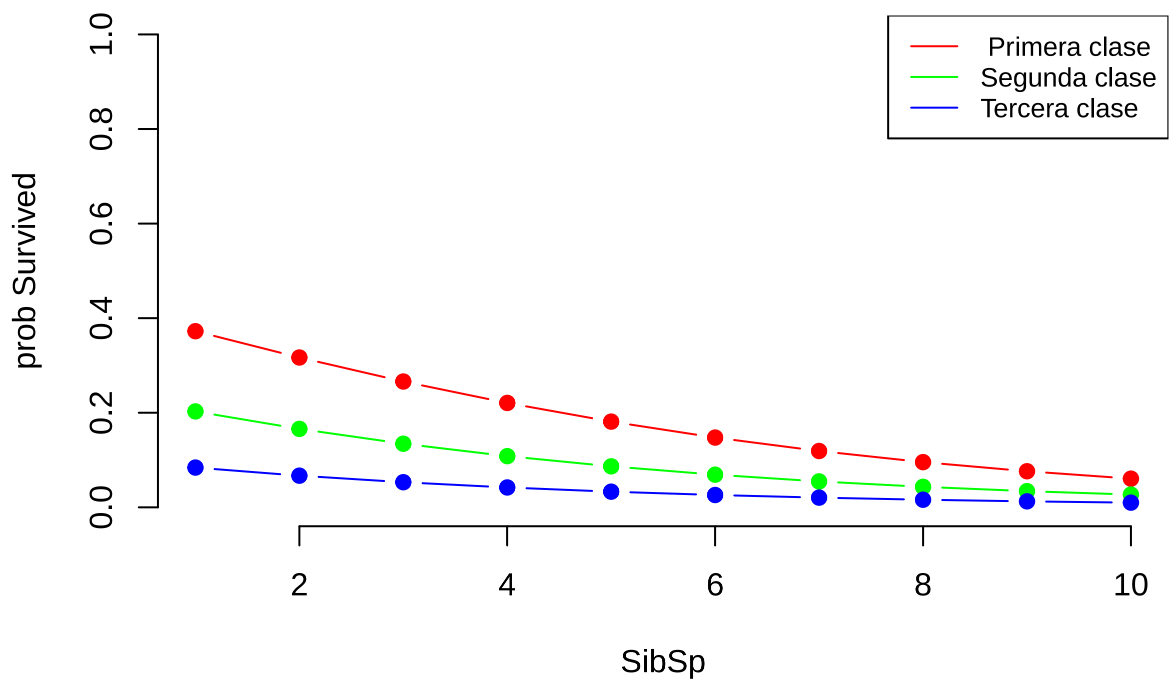
```r
lines(c(1, 2), prob_2, pch = 19, col = "green", type = "b")

lines(c(1, 2), prob_3, pch = 19, col = "blue", type = "b")

legend("topright", legend=c(" Primera clase", "Segunda clase", "Tercera clase"), col=c("red", "green",
```




Por último, solamente variaremos el SibSp. En el caso de los hombres:

```r
new_passengers_class_1 <- data.frame(
  Sex = rep("male", times = 10),
  Pclass = rep("1st", times = 10),
  SibSp = 1:10
)

new_passengers_class_2 <- data.frame(
  Sex = rep("male", times = 10),
  Pclass = rep(c("2nd"), times = 10),
  SibSp = 1:10
)

new_passengers_class_3 <- data.frame(
  Sex = rep("male", times = 10),
  Pclass = rep("3rd", times = 10),
  SibSp = 1:10
)
```

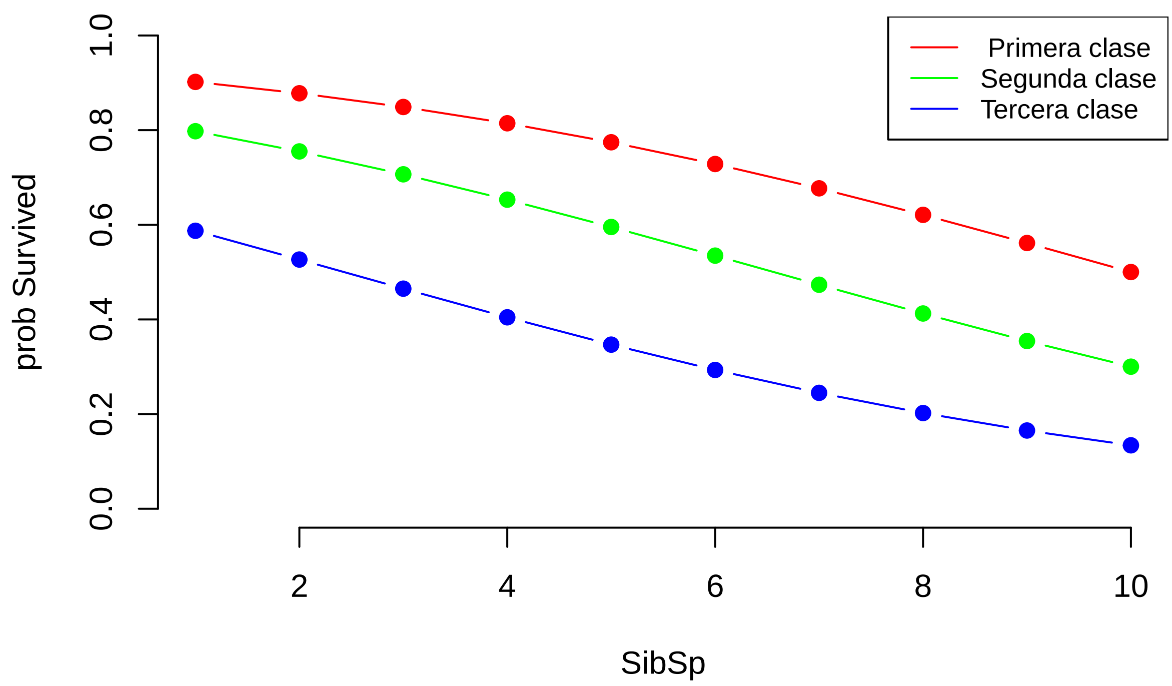
```r
prob_1 <- predict(model.logist3, newdata = new_passengers_class_1, type="response")

prob_2 <- predict(model.logist3, newdata = new_passengers_class_2, type="response")

prob_3 <- predict(model.logist3, newdata = new_passengers_class_3, type="response")

plot(c(1:10), prob_1, type = "b", frame = FALSE, pch = 19, col = "red", xlab = "SibSp", ylab = "prob Su

lines(c(1:10), prob_2, pch = 19, col = "green", type = "b")

lines(c(1:10), prob_3, pch = 19, col = "blue", type = "b")

legend("topright", legend=c(" Primera clase", "Segunda clase", "Tercera clase"), col=c("red", "green",
```




Y en el de las mujeres:

```r
new_passengers_class_1 <- data.frame(
  Sex = rep("female", times = 10),
  Pclass = rep("1st", times = 10),
  SibSp = 1:10
)

new_passengers_class_2 <- data.frame(
  Sex = rep("female", times = 10),
  Pclass = rep("2nd", times = 10),
  SibSp = 1:10
```

```
)

new_passengers_class_3 <- data.frame(
  Sex = rep("female", times = 10),
  Pclass = rep("3rd", times = 10),
  SibSp = 1:10
)

prob_1 <- predict(model.logist3, newdata = new_passengers_class_1, type="response")

prob_2 <- predict(model.logist3, newdata = new_passengers_class_2, type="response")

prob_3 <- predict(model.logist3, newdata = new_passengers_class_3, type="response")

plot(c(1:10), prob_1, type = "b", frame = FALSE, pch = 19, col = "red", xlab = "SibSp", ylab = "prob Su

lines(c(1:10), prob_2, pch = 19, col = "green", type = "b")

lines(c(1:10), prob_3, pch = 19, col = "blue", type = "b")

legend("topright", legend=c(" Primera clase", "Segunda clase", "Tercera clase"), col=c("red", "green",
```



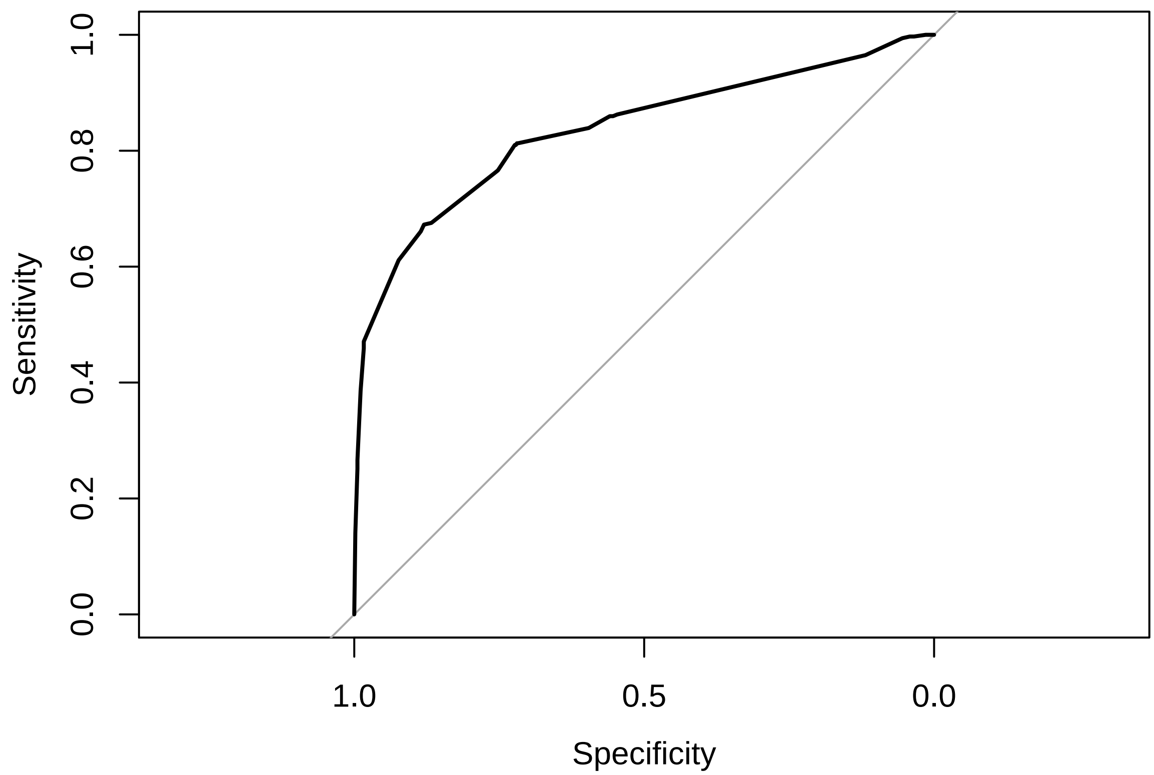

Vemos cómo se comporta nuestro modelo:

```
model=model.logist3
```

```r
prob=predict(model, ds, type="response")

r=roc(ds$Survived,prob, data=ds)
```

```
## Setting levels: control = Not, case = Yes

## Setting direction: controls < cases
```

```r
plot (r)
```




```r
auc(r)
```

```
## Area under the curve: 0.835
```

Vemos que el **área bajo la curva es de 0.8328, por lo que la capacidad de predicción de nuestro modelo es bastante buena**. Procedemos a calcular la sensibilidad y la especifidad.

```r
calculate_sensibility <- function(confusion_matrix){
  if(ncol(confusion_matrix) != 2) return(0)

  yes_yes <- confusion_matrix[2,2]
  yes_no <- confusion_matrix[1,2]

  sensibility <- yes_yes / (yes_yes + yes_no)

  return(sensibility)
}
```

```
calculate_specifity <- function(confusion_matrix){
  if(ncol(confusion_matrix) != 2) return(0)

  no_no <- confusion_matrix[1,1]
  no_yes <- confusion_matrix[2,1]

  specifity <- no_no / (no_no + no_yes)

  return(specifity)
}


calculate_global_accuracy <- function(confusion_matrix){
  if(ncol(confusion_matrix) != 2) return(0)

  yes_yes <- confusion_matrix[2,2]
  yes_no <- confusion_matrix[1,2]
  no_no <- confusion_matrix[1,1]
  no_yes <- confusion_matrix[2,1]

  ok_results <- yes_yes + no_no
  ko_results <- yes_no + no_yes

  ok_results / (ok_results + ko_results)
}


calculate_confusion_matrix <- function(model, data, real_values, threshold){
  predictions <- ifelse(predict(model, newdata = data, type="response")<threshold, "No","Yes")

  table(real_values, predictions, dnn = c("Valor Real","Valor Predicho"))
}
```

A continuación, observamos a ver cómo evoluciona la calidad (sensibilidad, especifidad y calidad total) cambiando el umbral según el cual aceptaremos que nuestro modelo predice si un viajero se salvó o no:

```
calculate_quality_params <- function(model, data, real_values, threshold){

confusion_matrix <- calculate_confusion_matrix(model, data, real_values, threshold)

specifity <- calculate_specifity(confusion_matrix)

sensibility <- calculate_sensibility(confusion_matrix)

global_accuracy <- calculate_global_accuracy(confusion_matrix)

list("threshold" = threshold, "confusion_matrix" = confusion_matrix, "specifity" = specifity, "sensibili
}

quality_params_06 <- calculate_quality_params(model.logist3, ds, ds$Survived, 0.6)

quality_params_07 <- calculate_quality_params(model.logist3, ds, ds$Survived, 0.7)

quality_params_08 <- calculate_quality_params(model.logist3, ds, ds$Survived, 0.8)
```

```r
quality_params_85 <- calculate_quality_params(model.logist3, ds, ds$Survived, 0.85)

quality_params_09 <- calculate_quality_params(model.logist3, ds, ds$Survived, 0.9)

quality_params_06
```

```
## $threshold
## [1] 0.6
##
## $confusion_matrix
##             Valor Predicho
## Valor Real   No Yes
##        Not 507  42
##        Yes 133 209
##
## $specifity
## [1] 0.7921875
##
## $sensibility
## [1] 0.8326693
##
## $`global accuracy`
## [1] 0.8035915
```

```r
quality_params_07
```

```
## $threshold
## [1] 0.7
##
## $confusion_matrix
##             Valor Predicho
## Valor Real   No Yes
##        Not 540   9
##        Yes 181 161
##
## $specifity
## [1] 0.7489598
##
## $sensibility
## [1] 0.9470588
##
## $`global accuracy`
## [1] 0.7867565
```

```r
quality_params_08
```

```
## $threshold
## [1] 0.8
##
## $confusion_matrix
##             Valor Predicho
## Valor Real   No Yes
##        Not 543   6
##        Yes 210 132
##
```

```
## $specifity
## [1] 0.7211155
##
## $sensibility
## [1] 0.9565217
##
## $`global accuracy`
## [1] 0.7575758
```

```
quality_params_85
```

```
## $threshold
## [1] 0.85
##
## $confusion_matrix
##            Valor Predicho
## Valor Real  No Yes
##        Not 546   3
##        Yes 253  89
##
## $specifity
## [1] 0.6833542
##
## $sensibility
## [1] 0.9673913
##
## $`global accuracy`
## [1] 0.7126824
```

```
quality_params_09
```

```
## $threshold
## [1] 0.9
##
## $confusion_matrix
##            Valor Predicho
## Valor Real  No Yes
##        Not 546   3
##        Yes 256  86
##
## $specifity
## [1] 0.680798
##
## $sensibility
## [1] 0.9662921
##
## $`global accuracy`
## [1] 0.7093154
```

Vemos que **con un umbral del 0.6, obtenemos una gran sensibilidad (83%) sin comprometer la calidad total (80%)** por lo que **la calidad de nuestro modelo es bastante aceptable**.

## 6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

En primer lugar nos hemos preguntado si los niños sobrevivieron más que los adultos, **comparando el atributo Age entre estas dos subpoblaciones**. Si bien la variable Age no sigue una distribución normal y no podemos explicar el comportamiento de la variable Survived a partir de ella, sí **hemos concluído, con un 95% de confianza, que los niños sobrevivieron mucho más que los adultos**. Asimismo, la supervivencia de los niños está mucho más dispersa que la de los adultos.

Posteriormente, hemos construído un modelo de regresión lineal logística que explica la variable Survived con bastante calidad. El modelo es el siguiente:

$$Survived = exp(3.43 - 2.74 * Sexmale - 0.93 * Pclass - 0.24 * SibSp)$$

A través del modelo mismo y de las gráficas de predicciones del mismo, hemos descubierto que:

- Aunque los niños sobreviviesen mucho más que los adultos, **no podemos establecer un modelo que explique la variable Survived con el atributo Age**.
- En general, **los hombres tienen muchas menos probabilidades de sobrevivir que las mujeres**.
- **La clase también tiene un papel fundamental**. Sin importar esposa o hermanos, **un hombre de tercera clase *a priori* tiene muy pocas probabilidades de haber sobrevivido**.
- Sorprendentemente, **la variable SibSp es la que más peso tiene**. **A partir de 6 hermanos / esposa un hombre, independientemente de su clase, tiene muy pocas probabilidades de sobrevivir**. Podemos observar también cómo **en las mujeres este efecto es menos acusado**, y que una mujer de primera clase, incluso yendo con muchos hermanos, sí tenía mucha más probabilidad de sobrevivir que un hombre.