Práctica 2: Limpieza y análisis de datos

Pedro Uceda Martínez, Pablo Campillo Sánchez

1 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 examinaremos qué posibles factores pudieron influir en la supervivencia de los pasajeros. Así, podremos, por ejemplo, ver si solamente la clase del billete, el género (mujeres) y la edad (niños) 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 los mismos y, posteriormente, se procede 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 procedemos a cargar el **dataset**, sin **factors**, para evitar tratar los nombres de los pasajeros como tales.

```
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. Tenemos las variables cuantitativas PassengerId, Survived, Pclass, Age, SibSp, Parch y Fare, todas tratadas como int o num. También están 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. Por cuestiones prácticas no la transformamos.

Para más claridad de los datos, procedemos a realizar las siguientes transformaciones: - 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 en vez de cadena.

```
ds$Embarked <- factor(ds$Embarked, levels=sort(c("C", "Q", "S")), labels = c("Cherbourg", "Queenstown",
ds$Sex <- factor(ds$Sex)
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        : 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:

```
summary(ds)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex           Age            SibSp           Parch
##  female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  male  :577   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
```

```
##                Median :28.00   Median :0.000   Median :0.0000
##                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
##     Ticket               Fare           Cabin                  Embarked
##  Length:891         Min.   :  0.00   Length:891         Cherbourg  :168
##  Class :character   1st Qu.:  7.91   Class :character   Queenstown : 77
##  Mode  :character   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.

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:

```
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.

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

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

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

```
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 10.

Ticket 1601:

```
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      3 56.4958       Southampton male  32     0     0
## 170   Ling, Mr. Lee      3 56.4958       Southampton male  28     0     0
## 510  Lang, Mr. Fang      3 56.4958       Southampton male  26     0     0
## 644 Foo, Mr. Choong      3 56.4958       Southampton male  NA     0     0
## 693    Lam, Mr. Ali      3 56.4958       Southampton male  NA     0     0
## 827    Lam, Mr. Len      3 56.4958       Southampton male  NA     0     0
## 839 Chip, Mr. Chang      3 56.4958       Southampton male  32     0     0
```

Ticket 347082:

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

```
##                                                      Name Pclass   Fare
## 14                              Andersson, Mr. Anders Johan      3 31.275
## 120                          Andersson, Miss. Ellis Anna Maria      3 31.275
## 542                        Andersson, Miss. Ingeborg Constanzia      3 31.275
## 543                           Andersson, Miss. Sigrid Elisabeth      3 31.275
## 611 Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren)      3 31.275
## 814                          Andersson, Miss. Ebba Iris Alfrida      3 31.275
## 851                    Andersson, Master. Sigvard Harald Elias      3 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:

```
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      3 69.55       Southampton   male  NA
## 181      Sage, Miss. Constance Gladys      3 69.55       Southampton female  NA
## 202               Sage, Mr. Frederick      3 69.55       Southampton   male  NA
## 325          Sage, Mr. George John Jr      3 69.55       Southampton   male  NA
## 793           Sage, Miss. Stella Anna      3 69.55       Southampton female  NA
## 847          Sage, Mr. Douglas Bullen      3 69.55       Southampton   male  NA
## 864 Sage, Miss. Dorothy Edith "Dolly"      3 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:

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

```
##                                                 Name Pclass Fare Cabin
## 64                              Skoog, Master. Harald      3 27.9
## 168 Skoog, Mrs. William (Anna Bernhardina Karlsson)      3 27.9
## 361                                Skoog, Mr. Wilhelm      3 27.9
## 635                                Skoog, Miss. Mabel      3 27.9
## 643                     Skoog, Miss. Margit Elizabeth      3 27.9
## 820                      Skoog, Master. Karl Thorsten      3 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:

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

```
##                                      Name Pclass    Fare Cabin    Embarked
## 51                Panula, Master. Juha Niilo      3 39.6875       Southampton
## 165             Panula, Master. Eino Viljami      3 39.6875       Southampton
## 267                Panula, Mr. Ernesti Arvid      3 39.6875       Southampton
## 639 Panula, Mrs. Juha (Maria Emilia Ojala)      3 39.6875       Southampton
## 687                 Panula, Mr. Jaako Arnold      3 39.6875       Southampton
## 825             Panula, Master. Urho Abraham      3 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:

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

```
##                                                 Name Pclass Fare Cabin
## 64                              Skoog, Master. Harald      3 27.9
## 168 Skoog, Mrs. William (Anna Bernhardina Karlsson)      3 27.9
## 361                                Skoog, Mr. Wilhelm      3 27.9
## 635                                Skoog, Miss. Mabel      3 27.9
## 643                     Skoog, Miss. Margit Elizabeth      3 27.9
## 820                      Skoog, Master. Karl Thorsten      3 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:

```
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      3 46.9       Southampton
## 72               Goodwin, Miss. Lillian Amy      3 46.9       Southampton
## 387         Goodwin, Master. Sidney Leonard      3 46.9       Southampton
## 481          Goodwin, Master. Harold Victor      3 46.9       Southampton
## 679 Goodwin, Mrs. Frederick (Augusta Tyler)      3 46.9       Southampton
## 684             Goodwin, Mr. Charles Edward      3 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:

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

```
##                                    Name Pclass   Fare Cabin   Embarked    Sex
## 17                  Rice, Master. Eugene      3 29.125       Queenstown   male
## 172                 Rice, Master. Arthur      3 29.125       Queenstown   male
## 279                   Rice, Master. Eric      3 29.125       Queenstown   male
## 788            Rice, Master. George Hugh      3 29.125       Queenstown   male
## 886 Rice, Mrs. William (Margaret Norton)      3 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:

```
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      2 73.5       Southampton male  21     0
## 121 Hickman, Mr. Stanley George      2 73.5       Southampton male  21     2
## 386   Davies, Mr. Charles Henry      2 73.5       Southampton male  18     0
## 656   Hickman, Mr. Leonard Mark      2 73.5       Southampton male  24     2
## 666          Hickman, Mr. Lewis      2 73.5       Southampton male  32     2
##     Parch
## 73      0
## 121     0
## 386     0
## 656     0
## 666     0
```

```
head(sort(distinct(ds, Ticket)$Ticket))
```

```
## [1] "110152" "110413" "110465" "110564" "110813" "111240"
```

2.2 Selección y creación de características

Los atributos PassengerId y Name no serán objeto de análisis.

Nótese que Cabin es susceptible de ser dividida en letra y número.

2.1 Carga de los datos y selección

2.2 Transformación de los datos

A continuación analizamos cada uno de los distintos atributos:

```
summary(ds)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex           Age            SibSp           Parch
##  female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  male  :577   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##               Median :28.00   Median :0.000   Median :0.0000
##               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
##     Ticket               Fare           Cabin                  Embarked
##  Length:891         Min.   :  0.00   Length:891         Cherbourg  :168
##  Class :character   1st Qu.:  7.91   Class :character   Queenstown : 77
##  Mode  :character   Median : 14.45   Mode  :character   Southampton:644
##                     Mean   : 32.20                      NA's       :  2
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

Vemos que los campos Age y Embarked tienen 177 y 2 valores nulos, respectivamente. Como no tiene sentido interpretarlos como 0 años o ningún puerto, sustituimos estos campos por la mediana para que afecten en la medida de lo posible al análisis.

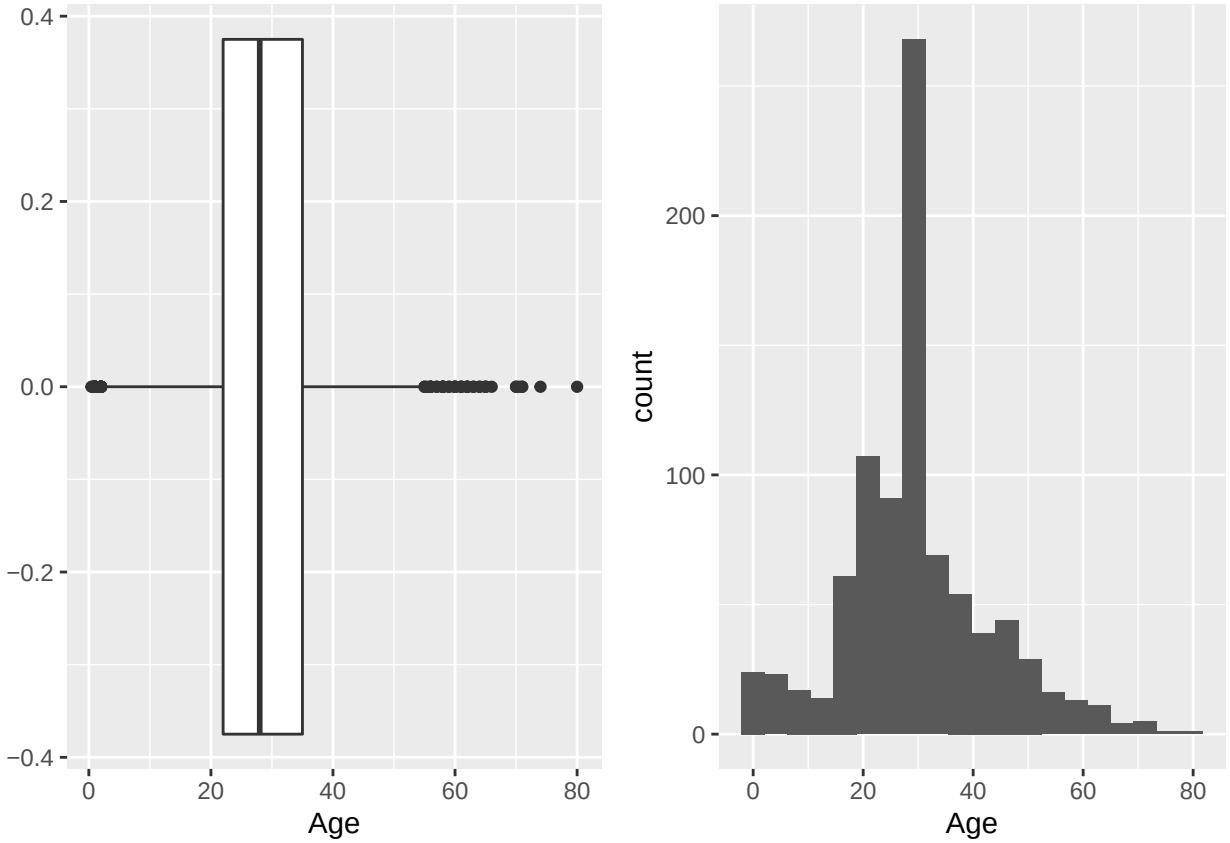
```
age_median <- median(ds$Age, na.rm = TRUE)

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

summary(ds)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
```

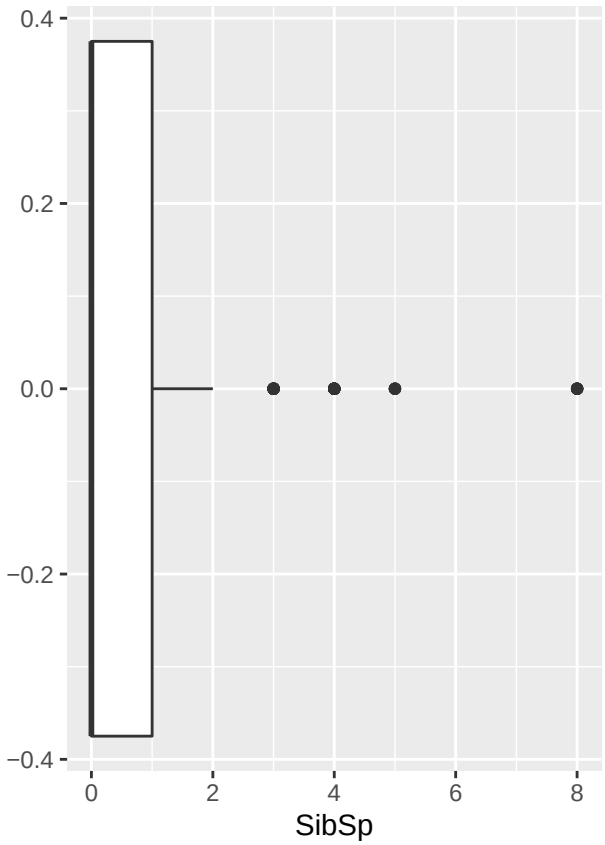
```
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##      Sex           Age             SibSp           Parch
##  female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  male  :577   1st Qu.:22.00   1st Qu.:0.000   1st Qu.:0.0000
##               Median :28.00   Median :0.000   Median :0.0000
##               Mean   :29.36   Mean   :0.523   Mean   :0.3816
##               3rd Qu.:35.00   3rd Qu.:1.000   3rd Qu.:0.0000
##               Max.   :80.00   Max.   :8.000   Max.   :6.0000
##     Ticket               Fare           Cabin                  Embarked
##  Length:891         Min.   :  0.00   Length:891         Cherbourg  :170
##  Class :character   1st Qu.:  7.91   Class :character   Queenstown : 77
##  Mode  :character   Median : 14.45   Mode  :character   Southampton:644
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
```

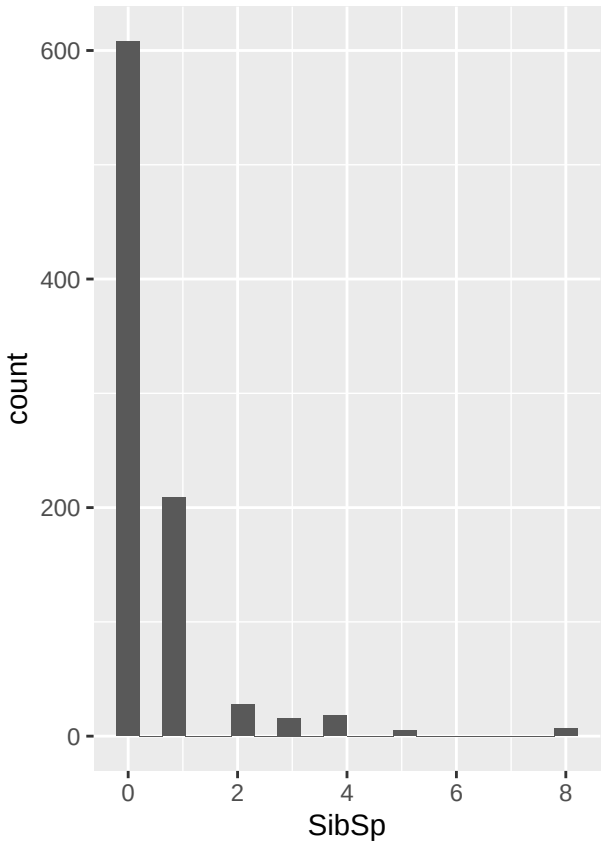
```
#Visualización de variables cuantitativas

#Age
gAge1 <- ggplot(ds, aes(x=Age)) + geom_boxplot()

gAge2 <- ggplot(ds, aes(x=Age)) + geom_histogram(bins=20)

#SibSp

gSibSp1 <- ggplot(ds, aes(x=SibSp)) + geom_boxplot()

gSibSp2 <- ggplot(ds, aes(x=SibSp)) + geom_histogram(bins=20)

#Parch

gParch1 <- ggplot(ds, aes(x=Parch)) + geom_boxplot()

gParch2 <- ggplot(ds, aes(x=Parch)) + geom_histogram(bins=20)


#Fare

gFare1 <- ggplot(ds, aes(x=Fare)) + geom_boxplot()

gFare2 <- ggplot(ds, aes(x=Fare)) + geom_histogram(bins=20)
```

```
grid.arrange(gAge1,gAge2,nrow=1)
```

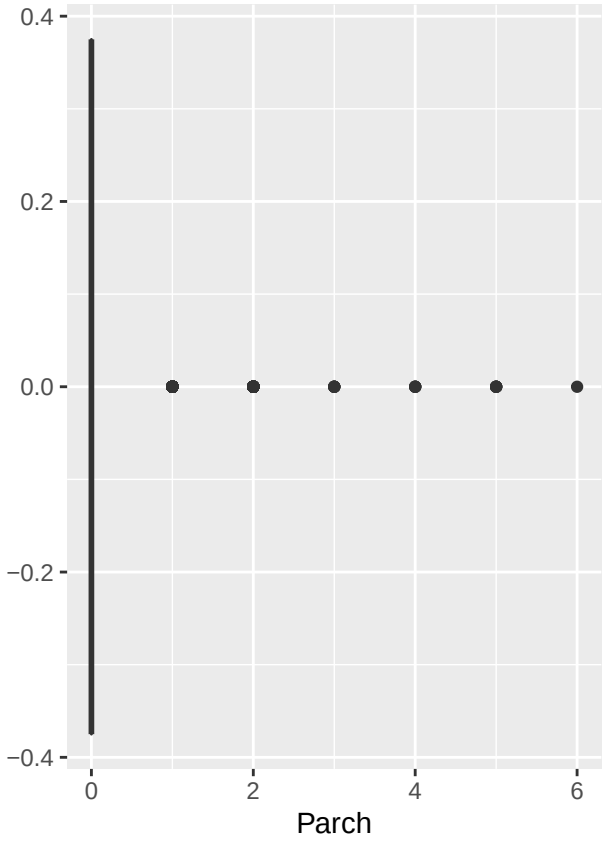


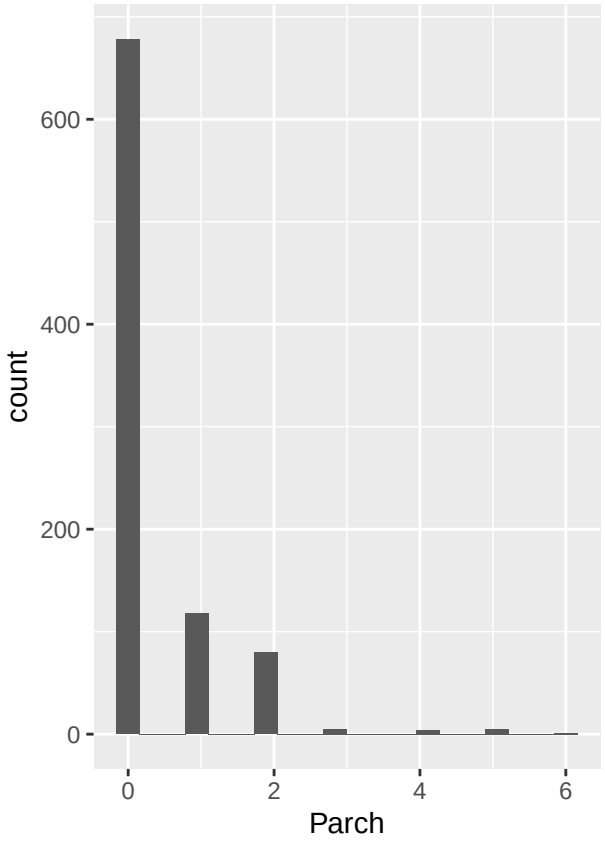

```
grid.arrange(gSibSp1,gSibSp2,nrow=1)
```



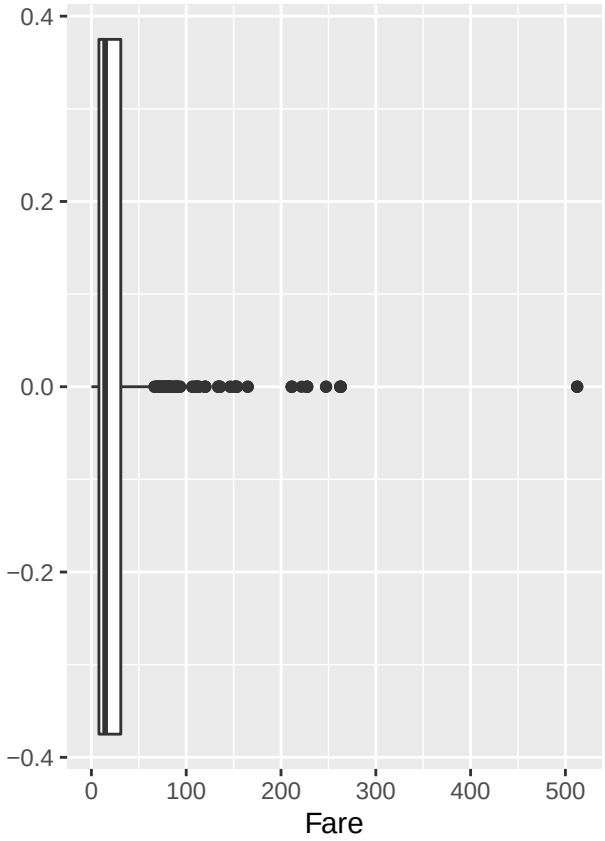


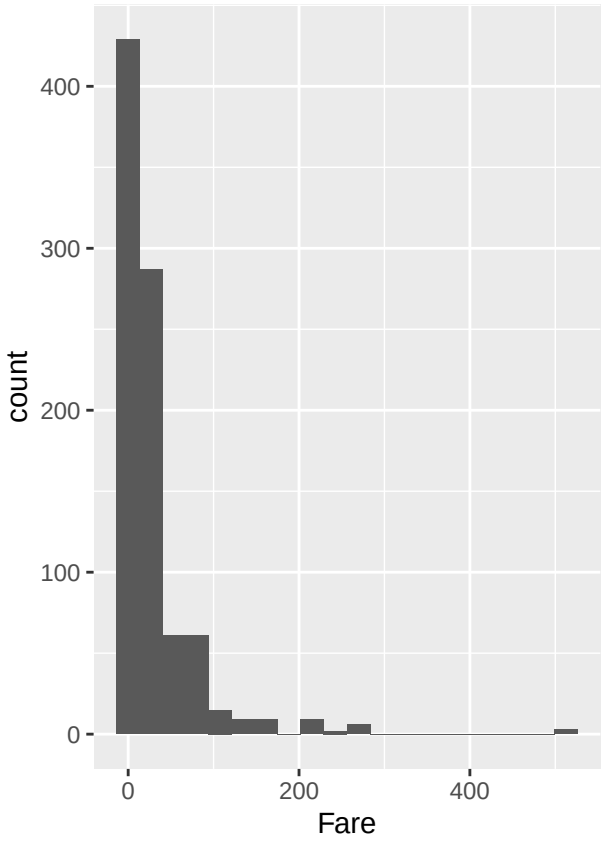

```
grid.arrange(gParch1,gParch2,nrow=1)
```






```
grid.arrange(gFare1,gFare2,nrow=1)
```






```
#Visualizacion de variables cuantitativas

#Survived
sumSurvived <- summarize( group_by(ds, Survived), n=length(Survived), Fare=mean(Fare))
```

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

```
gSurvived1 <- ggplot( sumSurvived, aes(x="", y=n, fill=Survived)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Survived")

#PClass and Survived
sumPClass <- summarize( group_by(ds, Pclass), n=length(Pclass), Survived=mean(Survived))
```

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

```
gPClass1 <- ggplot( sumPClass, aes(x="", y=n, fill=Pclass)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("PClass")

gPClass2 <- ds %>%
  group_by(Survived, Pclass) %>%
  tally() %>%
  group_by(Survived) %>%
  mutate(x = n / sum(n)) %>%
  ggplot() +
    geom_col(aes(
      x = factor(Pclass),
```

```
      y = x,
      fill = factor(Survived)
      ), position = "stack")

#Sex and Survived
sumSex <- summarize( group_by(ds, Sex), n=length(Sex), Survived=mean(Survived))
```

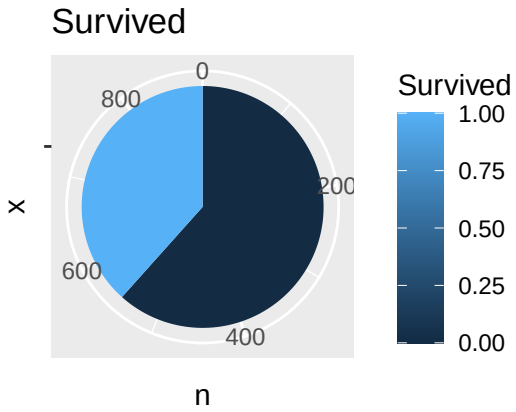
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
gSex1 <- ggplot( sumSex, aes(x="", y=n, fill=Sex)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Sex")

gSex2 <- ds %>%
  group_by(Survived, Sex) %>%
  tally() %>%
  group_by(Survived) %>%
  mutate(x = n / sum(n)) %>%
  ggplot() +
    geom_col(aes(
      x = factor(Sex),
      y = x,
      fill = factor(Survived)
      ), position = "stack")

#Embarked and Survived
sumEmbarked <- summarize( group_by(ds, Embarked), n=length(Embarked))
```

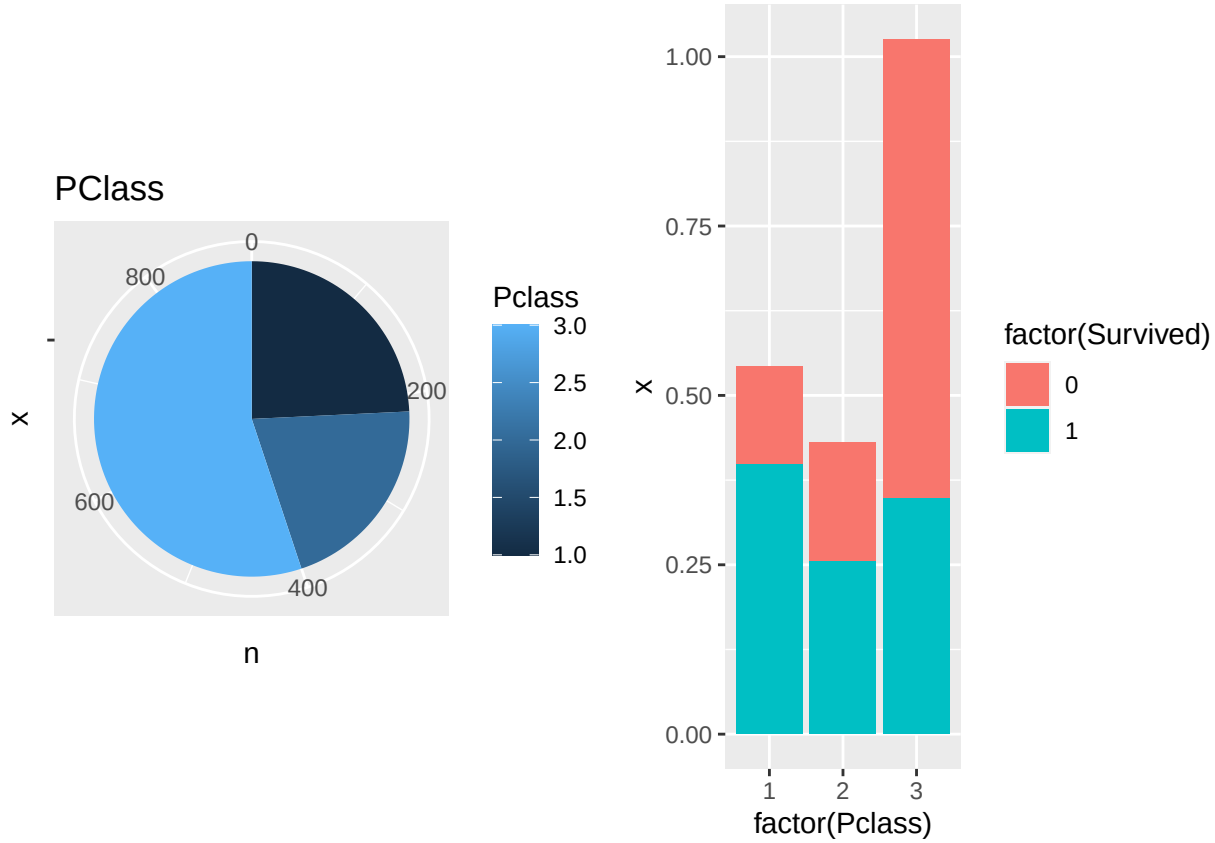
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
gEmbarked1 <- ggplot( sumEmbarked, aes(x="", y=n, fill=Embarked)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Embarked")


gEmbarked2 <- ds %>%
  group_by(Survived, Embarked) %>%
  tally() %>%
  group_by(Survived) %>%
  mutate(x = n / sum(n)) %>%
  ggplot() +
    geom_col(aes(
      x = factor(Embarked),
      y = x,
      fill = factor(Survived)
      ), position = "stack")
```

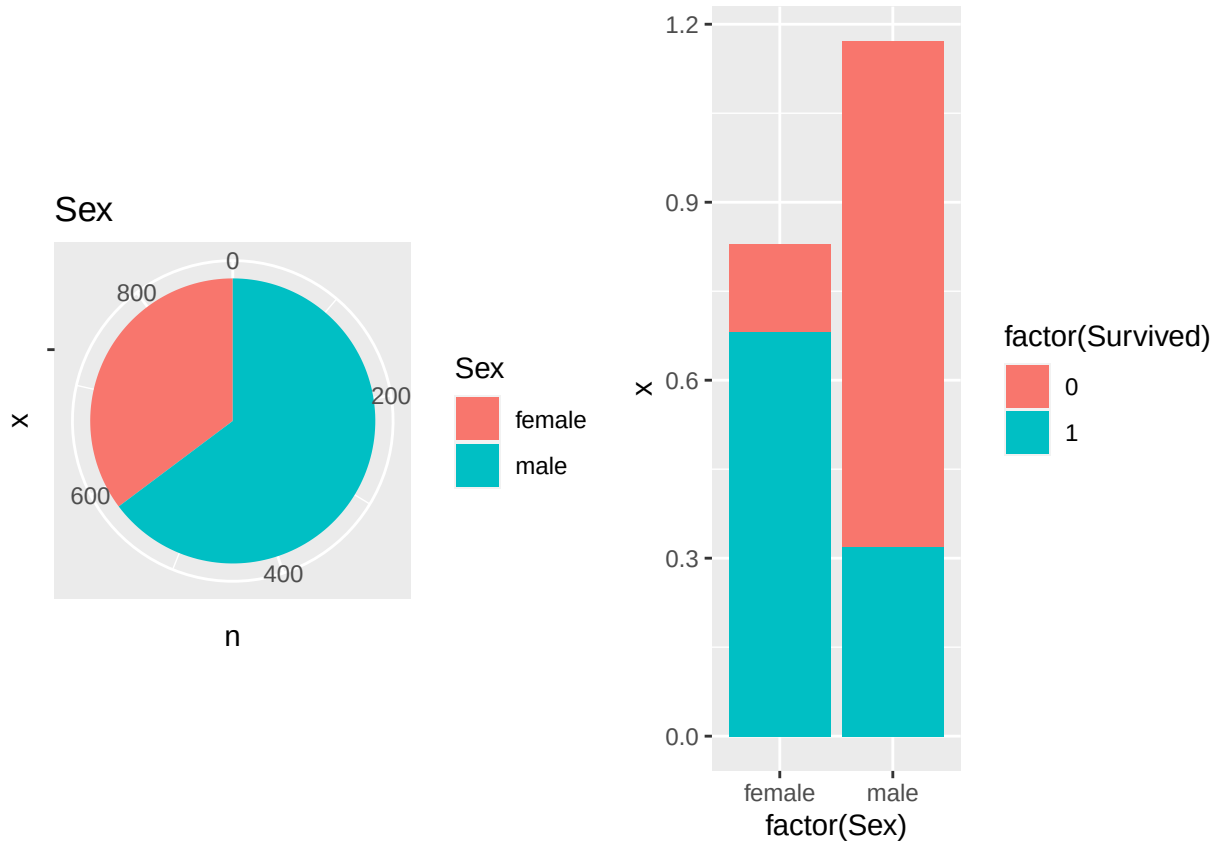
```
grid.arrange(gSurvived1, nrow=2)
```



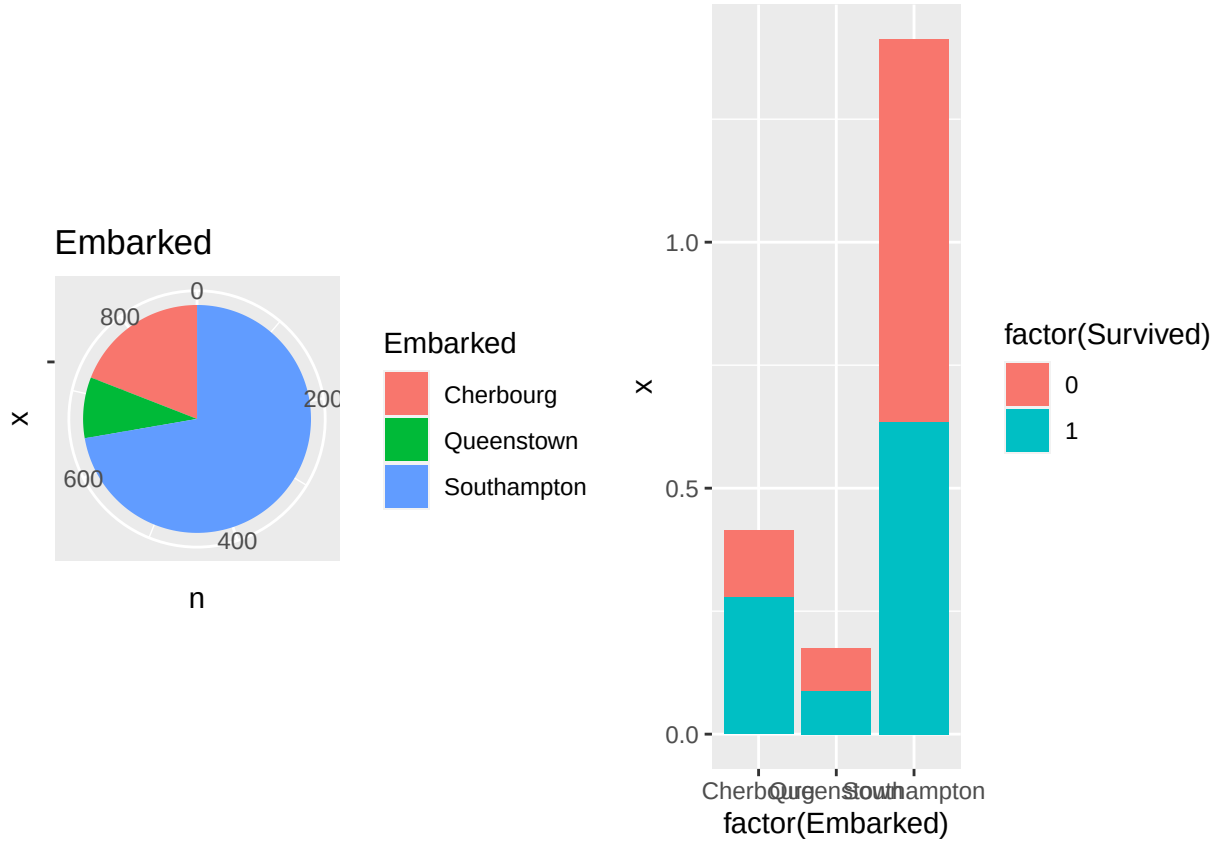

```
grid.arrange(gPClass1,gPClass2, nrow=1)
```




```
grid.arrange(gSex1, gSex2, nrow=1)
```




```
grid.arrange(gEmbarked1, gEmbarked2, nrow=1)
```



#2.3 Descipción estadística descriptiva

TODO: Describir cómo se distribuyen los datos y como podría saltar a la vista correlaciones. Da idea del ejercicio 4.

#3. Limpieza de datos

3.1 Elementos vacíos

TODO: En el ejercicio 1 se ha pintado el campo Age y el campo Embarked ya sin elementos vacíos. Traer aqui y pintar de nuevo, con un summary para demostrar que han desaparecido.

3.2 Identificación y tratamiento de valores extremos.

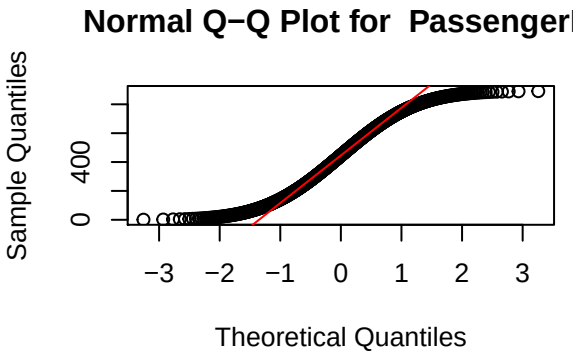
TODO: Explicar que hay valores extremos pero no podemos suponer que sean incorrectos (por ejemplo gente que tiene 8 hermanos o un billete que cuesta 500$). Poner ejemplos...

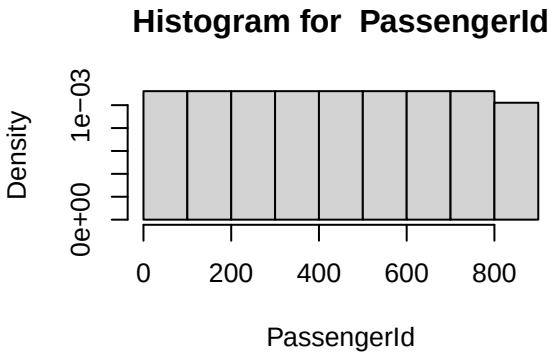
4. Análisis de los datos

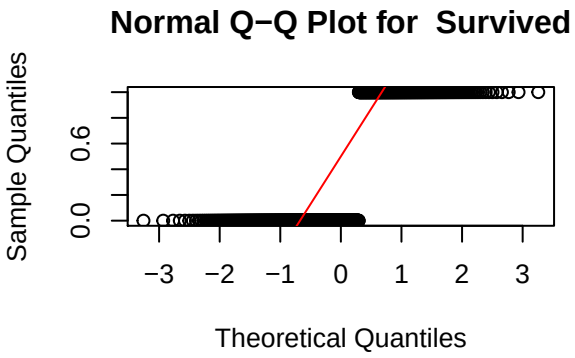
Antes de proceder a ver qué grupos de datos queremos normalizar, vamos a ver qué datos son normales y cuáles no, de manera gráfica...

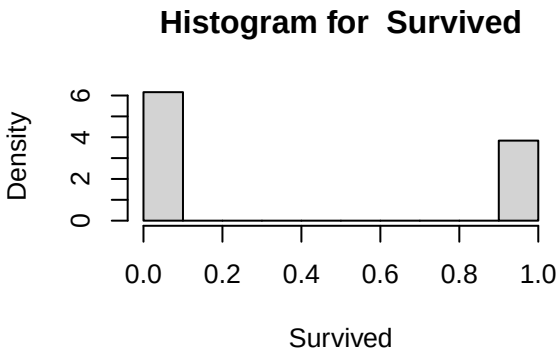
```
par(mfrow=c(2,2))
for(i in 1:ncol(ds)) {
  if (is.numeric(ds[,i])){
    qqnorm(ds[,i],main = paste("Normal Q-Q Plot for ",colnames(ds)[i]))
    qqline(ds[,i],col="red")
```

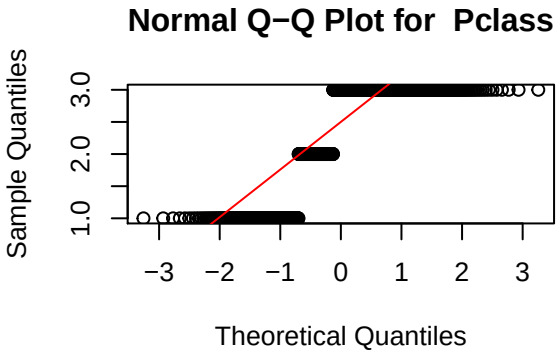
```
    hist(ds[,i],
      main=paste("Histogram for ", colnames(ds)[i]),
      xlab=colnames(ds)[i], freq = FALSE)
  }
}
```

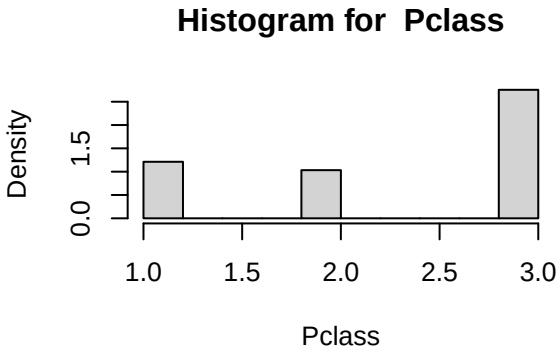


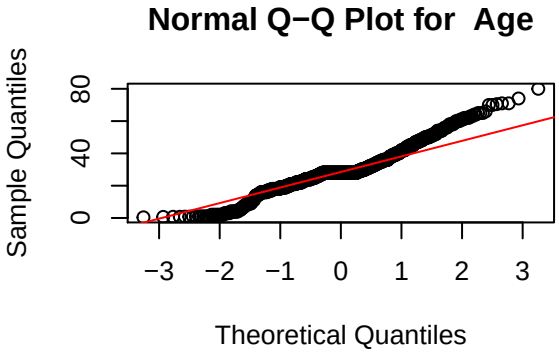


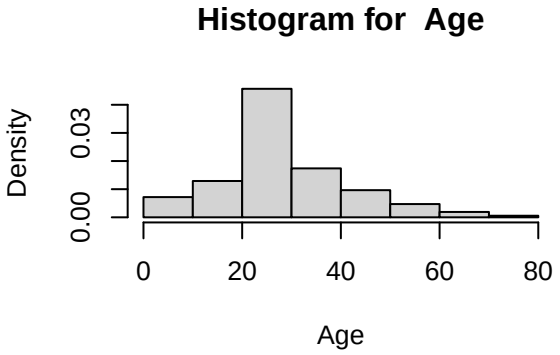


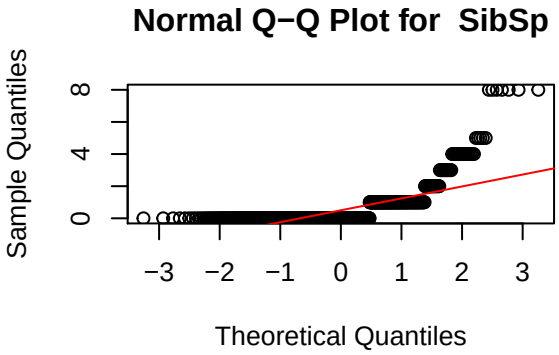


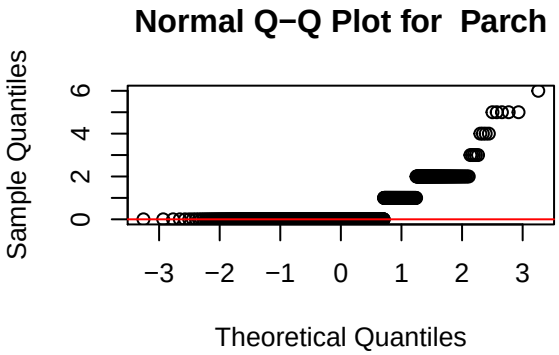

Normal Q−Q Plot for Pclass
Sample Quantiles
1.0
2.0
3.0
−3
−2
−1
0
1
2
3
Theoretical Quantiles

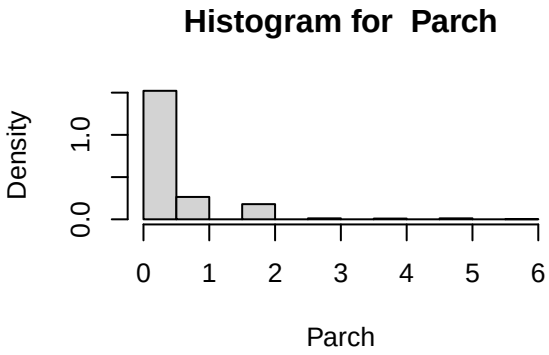

Histogram for Pclass
Density
0.0
1.5
1.0
1.5
2.0
2.5
3.0
Pclass


Normal Q−Q Plot for Age
Sample Quantiles
0
40
80
−3
−2
−1
0
1
2
3
Theoretical Quantiles

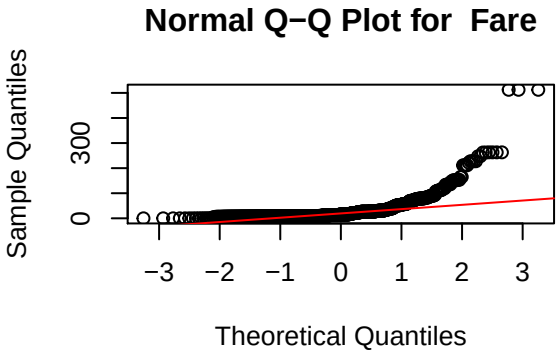

Histogram for Age
Density
0.00
0.03
0
20
40
60
80
Age

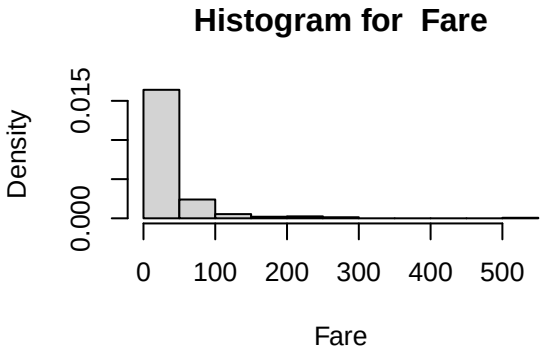


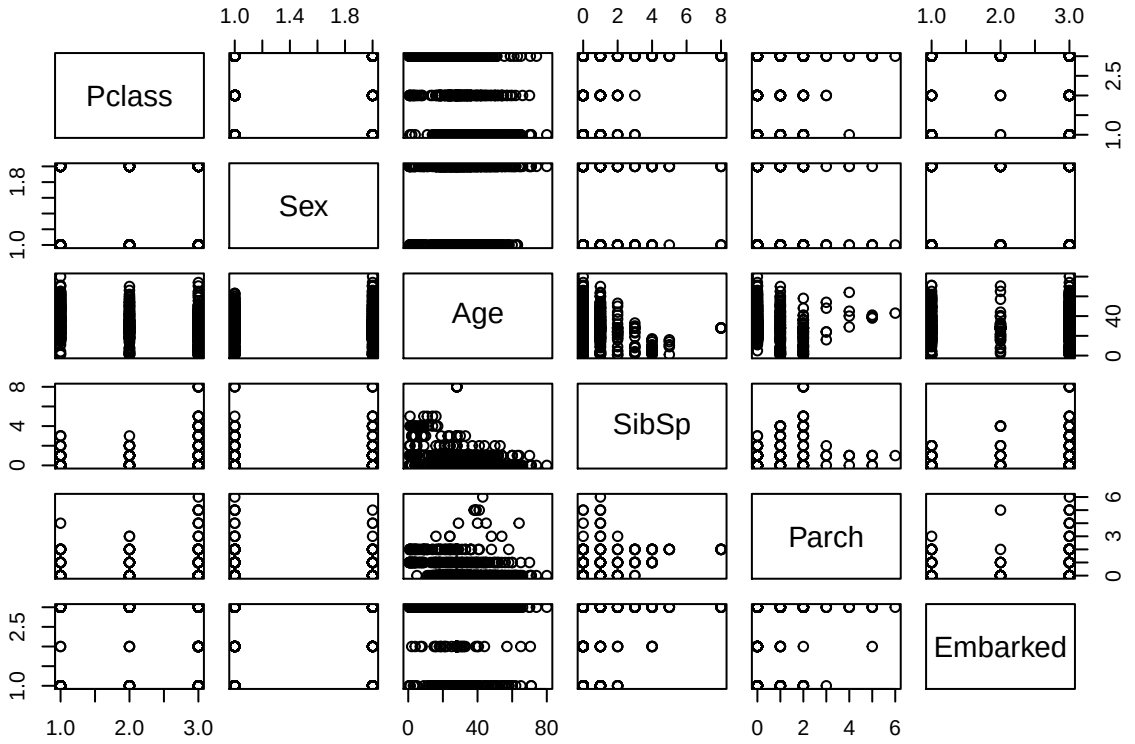





```
plot(ds[,c("Pclass", "Sex", "Age", "SibSp", "Parch", "Embarked")])
```


Normal Q−Q Plot for Fare
Sample Quantiles
0
300
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Histogram for Fare
Density
0.000
0.015
0
100
200
300
400
500
Fare



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:

```
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:

```
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:

```
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**.

```
  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
## -2.2030  -0.7036  -0.4519   0.6719   2.1599
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.2946     0.2974  11.077   <2e-16 ***
## Sexmale      -2.6434     0.1838 -14.380   <2e-16 ***
## Pclass       -0.9606     0.1061  -9.057   <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: 1186.7  on 890  degrees of freedom
## Residual deviance:  827.2  on 888  degrees of freedom
## AIC: 833.2
##
## 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
## -2.2689  -0.6735  -0.4747   0.6189   2.5148
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.43357    0.30543  11.242  < 2e-16 ***
## Sexmale     -2.74314    0.19027 -14.417  < 2e-16 ***
## Pclass      -0.93896    0.10647  -8.819  < 2e-16 ***
## SibSp       -0.24812    0.09453  -2.625  0.00867 **
## ---
## 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:  819.32  on 887  degrees of freedom
## AIC: 827.32
##
## Number of Fisher Scoring iterations: 4
```

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

Probamos a incorportar del mismo modo la variable Parch:

```
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
## -2.2794  -0.6849  -0.4761   0.6117   2.5292
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.45961    0.31139  11.110   <2e-16 ***
## Sexmale     -2.76236    0.19534 -14.142   <2e-16 ***
## Pclass      -0.93916    0.10653  -8.816   <2e-16 ***
## SibSp       -0.23402    0.09919  -2.359   0.0183 *
## Parch       -0.05026    0.11041  -0.455   0.6490
## ---
## 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:  819.11  on 886  degrees of freedom
## AIC: 829.11
##
## 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:

```
model.logist5 = glm(formula = Survived ~ Sex + Pclass + SibSp + Fare, family=binomial(link=logit), data

summary(model.logist5)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + SibSp + Fare, family = binomial(link = logit),
##     data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2587  -0.6607  -0.4788   0.6394   2.5244
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.140233   0.372913   8.421  < 2e-16 ***
## Sexmale     -2.727160   0.190773 -14.295  < 2e-16 ***
## Pclass      -0.847724   0.125523  -6.754 1.44e-11 ***
## SibSp       -0.277048   0.097580  -2.839  0.00452 **
## Fare         0.002997   0.002245   1.335  0.18185
## ---
## 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:  817.36  on 886  degrees of freedom
## AIC: 827.36
##
## Number of Fisher Scoring iterations: 5
```

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:

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

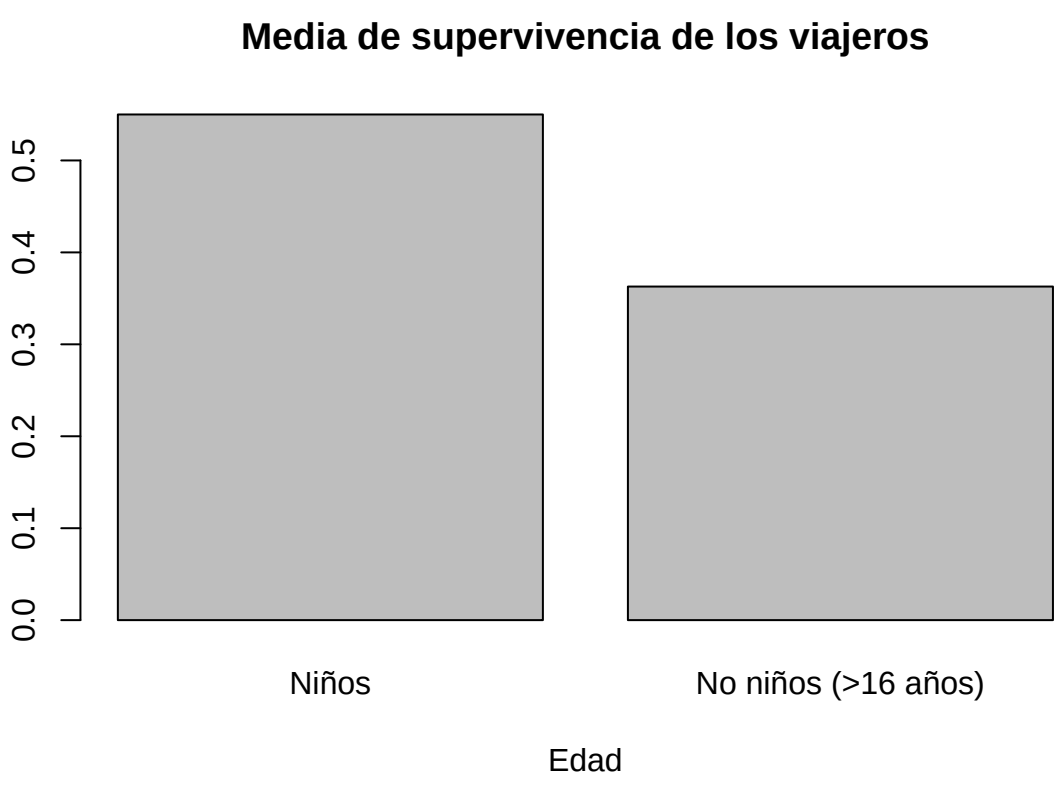
5. Representación de los resultados a partir de tablas y gráficas

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:

```
#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)
```

```r
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
```



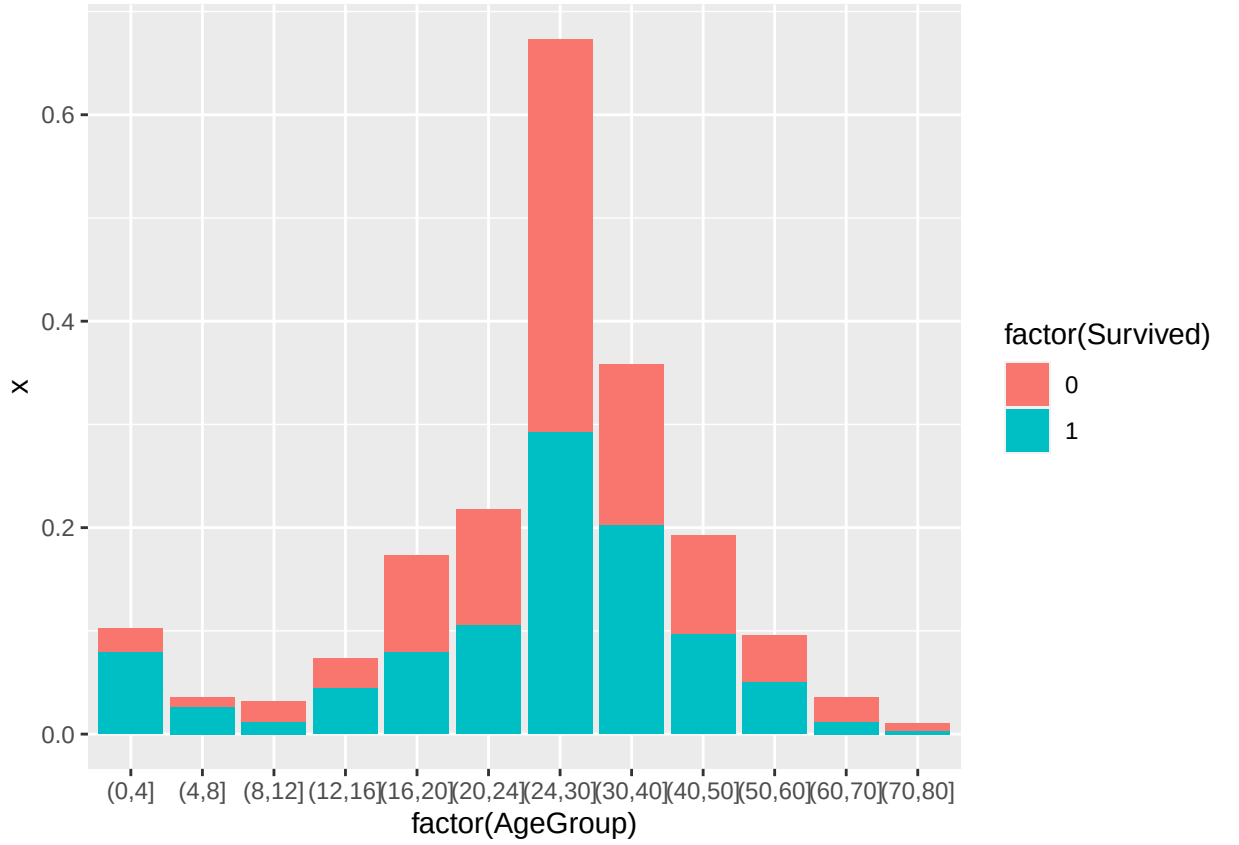
Podemos ver también **cómo se distribuye la supervivencia, agrupando los pasajeros por edades**:

```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() +
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:

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

```
## (Intercept)     Sexmale      Pclass       SibSp
## 30.98709984  0.06436822  0.39103409  0.78026533
```

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

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

```
##                   2.5 %      97.5 %
## (Intercept) 17.28579164 57.30688127
## Sexmale      0.04395459  0.09274774
## Pclass       0.31629329  0.48037134
## SibSp        0.64128351  0.93071700
```

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:

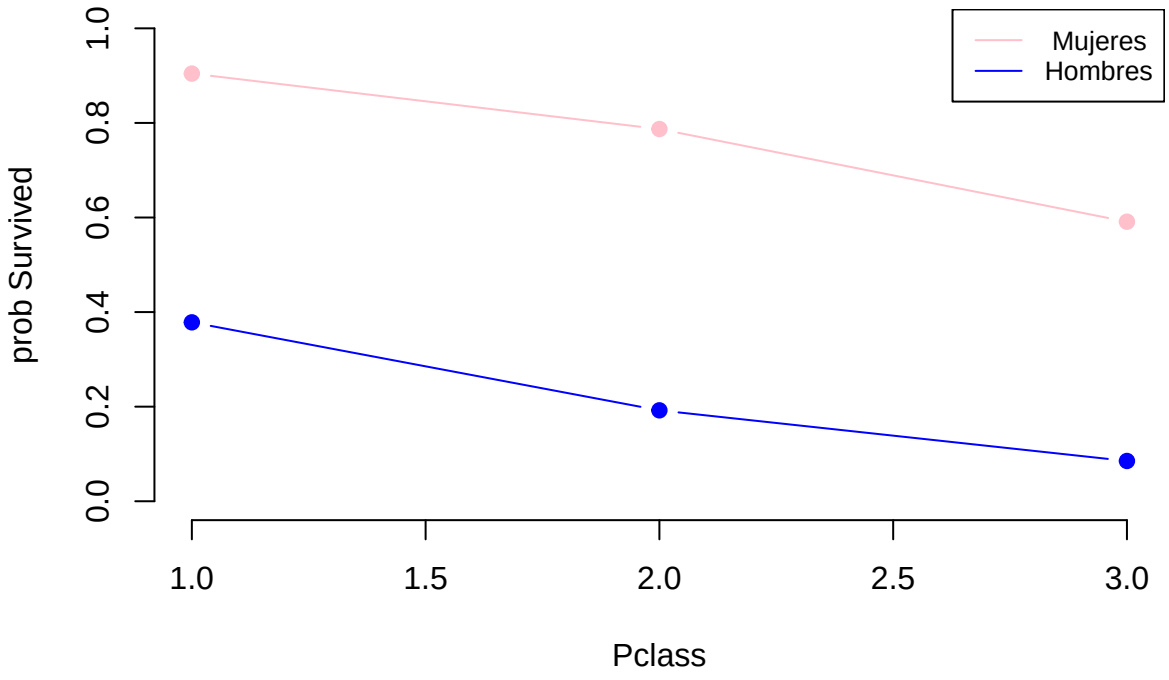
```
#Males
new_passengers_male <- data.frame(
  Sex = rep("male", times = 3),
  Pclass = c(1,2,3),
  SibSp = c(1,1,1)
)

#Females
new_passengers_female <- data.frame(
  Sex = rep("female", times = 3),
  Pclass = c(1,2,3),
  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
## 0.37832918 0.19222661 0.08513275
```

```
prob_females
```

```
##         1         2         3
## 0.9043473 0.7870993 0.5911129
```

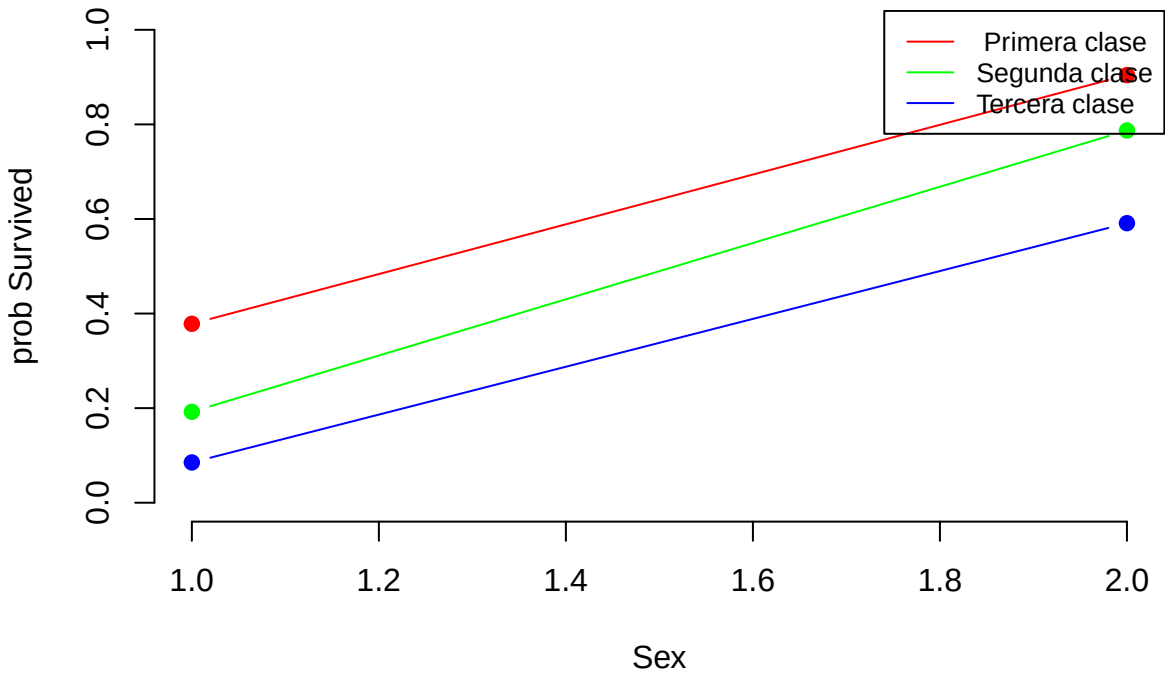
```
plot(c(1,2,3), prob_females, type = "b", frame = FALSE, pch = 19, col = "pink", xlab = "Pclass", ylab =

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(1,1),
  SibSp = c(1,1)
)

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

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

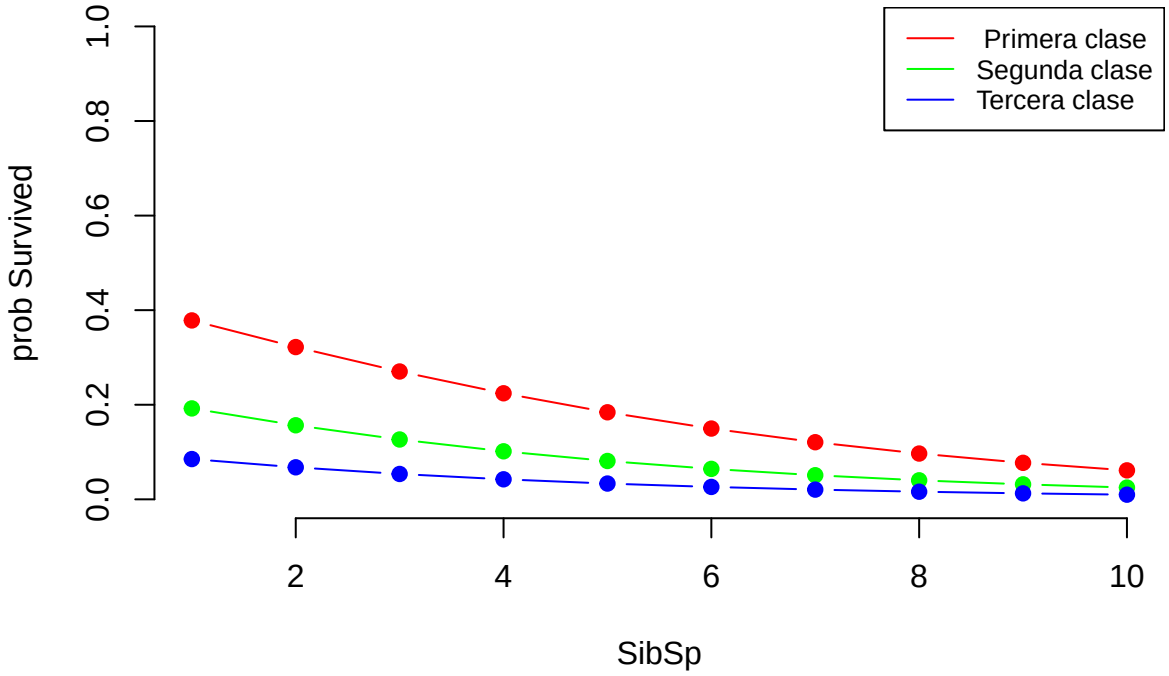
```r
lines(c(1, 2), prob_2, pch = 19, col = "green", type = "b")

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(1, times = 10),
  SibSp = 1:10
)

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

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

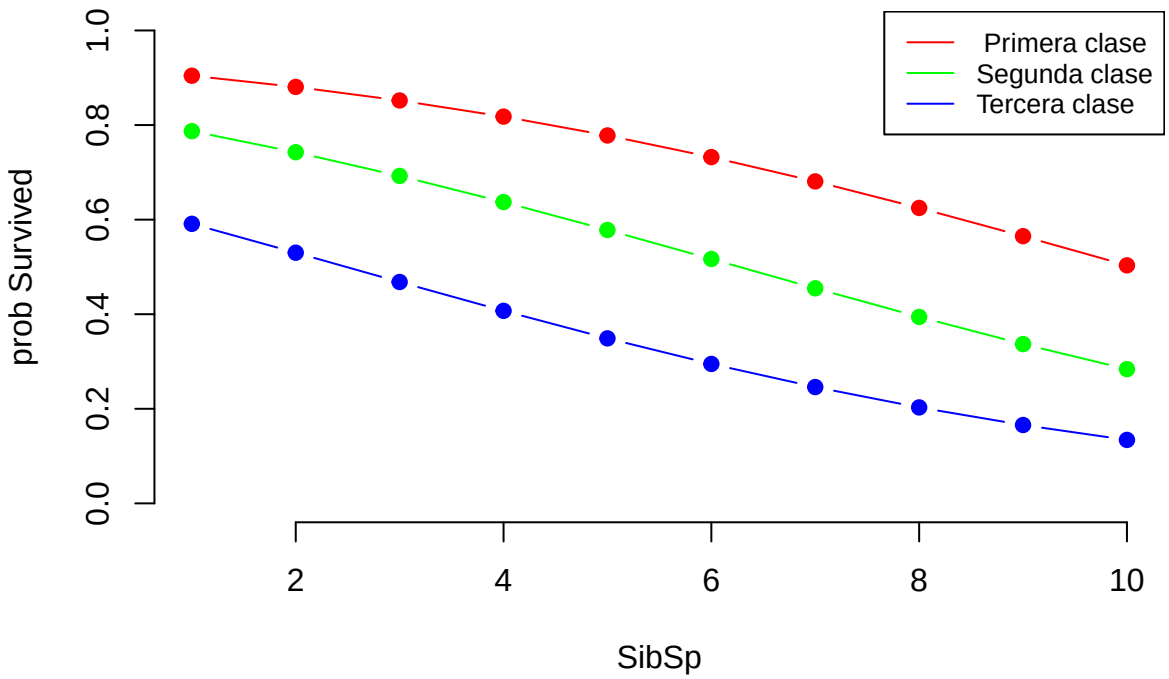
```r
prob_1 <- predict(model.logist3, newdata = new_passengers_class_1, type="response")

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(1, times = 10),
  SibSp = 1:10
)

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

```
)

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

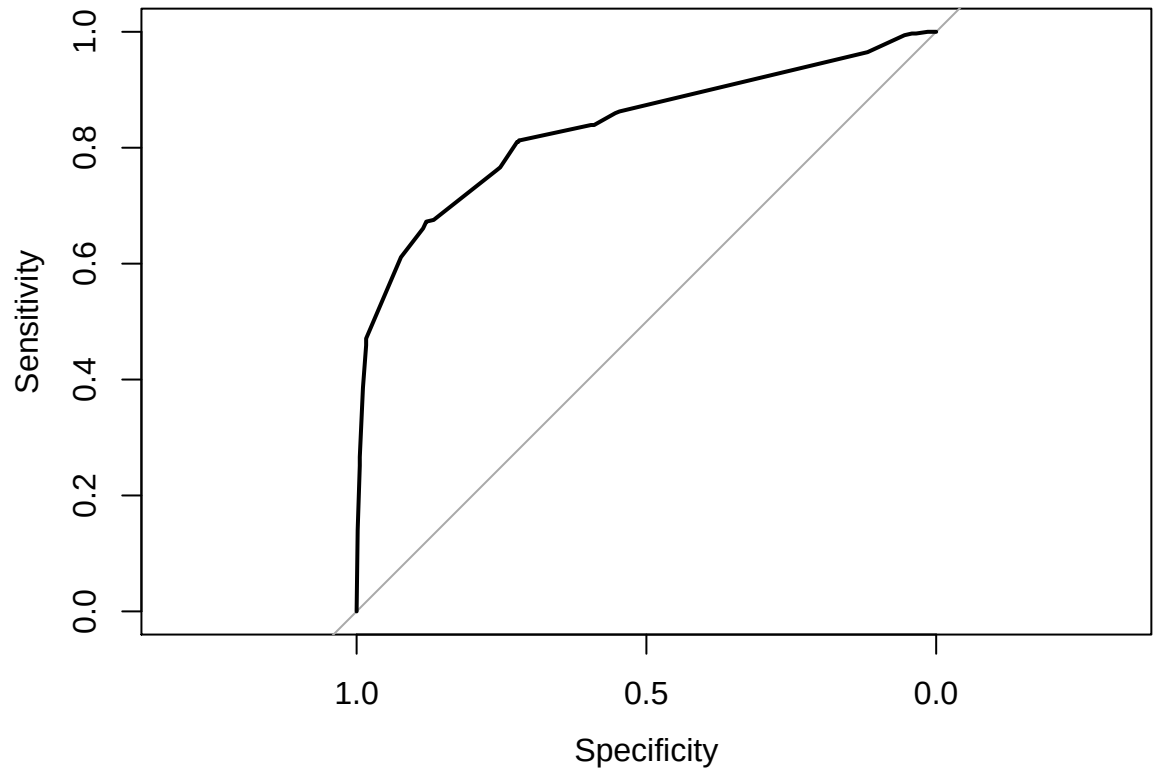
```
model=model.logist3
```

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

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

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot (r)
```




```
auc(r)
```

```
## Area under the curve: 0.8348
```

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.

```
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
##          0 507  42
##          1 133 209
##
## $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
##          0 540   9
##          1 182 160
##
## $specifity
## [1] 0.7479224
##
## $sensibility
## [1] 0.9467456
##
## $`global accuracy`
## [1] 0.7856341
```

```r
quality_params_08
```

```
## $threshold
## [1] 0.8
##
## $confusion_matrix
##           Valor Predicho
## Valor Real  No Yes
##          0 543   6
##          1 210 132
##
```

```
## $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
##          0 546   3
##          1 251  91
##
## $specifity
## [1] 0.685069
##
## $sensibility
## [1] 0.9680851
##
## $`global accuracy`
## [1] 0.714927
```

```
quality_params_09
```

```
## $threshold
## [1] 0.9
##
## $confusion_matrix
##           Valor Predicho
## Valor Real  No Yes
##          0 546   3
##          1 256  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.