Tipología y ciclo de vida de los datos

PRÁCTICA 2: Limpieza y análisis de datos

Autores: Eleazar Morales Díaz y Susana Vila Melero

5/6/2021

1. Descripción del dataset.

¿Qué grupo tiene mayor probabilidad de sobrevivir? ¿Hay relación entre categoría del billete y el pais o puerto de embarque?

Hemos elegido como dataset para realizar nuestra práctica el dataset de "Titanic", ya que nos permite realizar tareas predictivas sobre la variable **Survived**. En esa línea nuestro objetivo será analizar qué subconjunto de personas tendría mayor probabilidad de sobrevivir en el Titanic, a partir de los datos contenidos en el conjunto de datos. Estudiaremos también qué relación hay entre la categoría del billete y el país/puerto de embarque.

El dataset consta de 2.207 registros y 11 variables que se describen a continuación:

- **name**: nombre del pasajero (string).
- **gender**: información respecto al género del pasajero (factor con dos niveles).
- **age**: la edad del pasajero el día del naufragio. La edad de los bebés(menores de 12 meses) se proporciona como una fracción de un año (valor numérico).
- **class**: la clase para los pasajeros o el tipo de servicio para los miembros de la tripulación (factor).
- **embarked**: lugar de embarque del pasajero (factor).
- **country**: lugar de procedencia del pasajero (factor).
- **ticketno**: número de pasaje de los pasajeros, NA en el caso de ser miembros de la tripulación (valor numérico).
- **fare**: Precio del pasaje, NA para miembros de la tripulación, músicos y empleados de la compañía naviera (valor numérico).
- **sibsp**: número de esposas/hermanos a bordo, tomado del dataset Vanderbild (factor ordenado).
- **parch**: número de padres/hijos a bordo, tomado del dataset Vanderbild (factor ordenado).
- **survived**: información respecto a si el pasajero sobrevivió o no al naufragio (factor con dos niveles).

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

El primer paso será cargar las librerias y el dataset con el que vamos a trabajar.Una vez cargado, analizaremos su estructura y el tipo de variable y lo adecuaremos a nuestro estudio.

```
# Cargamos las librerías
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(RColorBrewer)
library(scales)
library(stats)
theme_set(theme_bw())

# Cargamos el dataset
ds <- read.csv("./data/titanic.csv", header=TRUE, fileEncoding="UTF-8")

#Hacemos una primera inspección
str(ds)
```

```
## 'data.frame':    2207 obs. of  11 variables:
##  $ name    : chr  "Abbing, Mr. Anthony" "Abbott, Mr. Eugene Joseph" "Abbott, Mr. Rossmore Edward" "Al
##  $ gender  : chr  "male" "male" "male" "female" ...
##  $ age     : num  42 13 16 39 16 25 30 28 27 20 ...
##  $ class   : chr  "3rd" "3rd" "3rd" "3rd" ...
##  $ embarked: chr  "S" "S" "S" "S" ...
##  $ country : chr  "United States" "United States" "United States" "England" ...
##  $ ticketno: int  5547 2673 2673 2673 348125 348122 3381 3381 2699 3101284 ...
##  $ fare    : num  7.11 20.05 20.05 20.05 7.13 ...
##  $ sibsp   : int  0 0 1 1 0 0 1 1 0 0 ...
##  $ parch   : int  0 2 1 1 0 0 0 0 0 0 ...
##  $ survived: chr  "no" "no" "no" "yes" ...
```

```
summary(ds)
```

```
##      name              gender               age              class
##  Length:2207        Length:2207        Min.   : 0.1667   Length:2207
##  Class :character   Class :character   1st Qu.:22.0000   Class :character
##  Mode  :character   Mode  :character   Median :29.0000   Mode  :character
##                                        Mean   :30.4367
##                                        3rd Qu.:38.0000
##                                        Max.   :74.0000
##                                        NA's   :2
##    embarked           country             ticketno            fare
##  Length:2207        Length:2207        Min.   :      2   Min.   :  3.030
##  Class :character   Class :character   1st Qu.:  14262   1st Qu.:  7.181
##  Mode  :character   Mode  :character   Median : 111426   Median : 14.090
##                                        Mean   : 284216   Mean   : 33.405
##                                        3rd Qu.: 347077   3rd Qu.: 31.061
##                                        Max.   :3101317   Max.   :512.061
##                                        NA's   :891       NA's   :916
##      sibsp            parch          survived
##  Min.   :0.0000   Min.   :0.0000   Length:2207
```

```
##  1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :0.4996   Mean   :0.3856
##  3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :8.0000   Max.   :9.0000
##  NA's   :900      NA's   :900
```

```
#Convertimos las variables carácter a variables factor
ds$gender = as.factor(ds$gender)
ds$class = as.factor(ds$class)
ds$embarked = as.factor(ds$embarked)
ds$survived = as.factor(ds$survived)
str(ds)
```

```
## 'data.frame':    2207 obs. of  11 variables:
##  $ name    : chr  "Abbing, Mr. Anthony" "Abbott, Mr. Eugene Joseph" "Abbott, Mr. Rossmore Edward" "Al
##  $ gender  : Factor w/ 2 levels "female","male": 2 2 2 1 1 2 2 1 2 2 ...
##  $ age     : num  42 13 16 39 16 25 30 28 27 20 ...
##  $ class   : Factor w/ 7 levels "1st","2nd","3rd",..: 3 3 3 3 3 3 2 2 3 3 ...
##  $ embarked: Factor w/ 4 levels "B","C","Q","S": 4 4 4 4 4 4 2 2 2 4 ...
##  $ country : chr  "United States" "United States" "United States" "England" ...
##  $ ticketno: int  5547 2673 2673 2673 348125 348122 3381 3381 2699 3101284 ...
##  $ fare    : num  7.11 20.05 20.05 20.05 7.13 ...
##  $ sibsp   : int  0 0 1 1 0 0 1 1 0 0 ...
##  $ parch   : int  0 2 1 1 0 0 0 0 0 0 ...
##  $ survived: Factor w/ 2 levels "no","yes": 1 1 1 2 2 2 1 2 2 2 ...
```

Ya tenemos el dataset cargado y con las variables transformadas para poder trabajar con él.

3. Limpieza de datos

3.1. Identificación y tratamiento de ceros o elementos vacíos.

Vamos a contar el total de **NAs** de nuestro dataset.

```
sum(is.na(ds))
```

```
## [1] 3690
```

Pero ¿cuál es el desglose de estos valores ausentes por variable?

```
colSums(is.na(ds))
```

```
##     name   gender      age    class embarked  country ticketno     fare
##        0        0        2        0        0       81      891      916
##    sibsp    parch survived
##      900      900        0
```

Esta información también la podemos obtener mediante `summary()`. El comando `summary(ds)` aplicado a nuestro dataset nos informa de que tenemos valores **NA** en las variables `age` (2 valores), `country` (81 valores), `ticketno` (891 valores), `fare` (916 valores), `sibsp` (900 valores) y `parch` (900 valores). Para cada

una de las variables será necesario detectar dichos valores ausentes y la toma de una medida para o bien sustituir el registro, omitirlo, o marcarlo como ausente en el conjunto de alguna forma. Este procedimiento se le conoce habitualmente como **imputación de valores**.

Como en edad sólo tenemos 2 valores ausentes, sustituir por el valor de la media.

```
index_is_na_age <- which(is.na(ds$age))

ds[c(index_is_na_age),]
```

```
##                       name gender age class embarked  country ticketno   fare
## 440 Gheorgheff, Mr. Stanio   male  NA   3rd        C Bulgaria   349254 7.1711
## 678    Kraeff, Mr. Theodor   male  NA   3rd        C Bulgaria   349253 7.1711
##     sibsp parch survived
## 440     0     0       no
## 678     0     0       no
```

Se observa que los dos registros son de Bulgaria, así que tiene más sentido aplicar la media de edad de las personas cuyo `country` sea el mismo.

```
which(ds$country=='Bulgaria') # para detectar todos aquellos que sean de Bulgaria
```

```
##  [1]   47   96  247  255  440  590  625  678  772  776  812  820  854  972  973
## [16] 1152 1165 1205 1211
```

```
as.integer(mean(ds[c(which( ds$country=='Bulgaria' )) , 'age'], na.rm = TRUE)) # media de los de Bulgar
```

```
## [1] 25
```

Realizamos la sustitución

```
ds[c(index_is_na_age),]$age <- as.integer(mean(ds[c(which( ds$country=='Bulgaria' )) , 'age'], na.rm = 

ds[c(index_is_na_age),]
```

```
##                       name gender age class embarked  country ticketno   fare
## 440 Gheorgheff, Mr. Stanio   male  25   3rd        C Bulgaria   349254 7.1711
## 678    Kraeff, Mr. Theodor   male  25   3rd        C Bulgaria   349253 7.1711
##     sibsp parch survived
## 440     0     0       no
## 678     0     0       no
```

Para la variable **country**, lugar de procedencia del pasajero se procede a sustituir los NAs por el valor que más se repite en el dataset, de esa forma reducimos el error.

```
names(sort(table(ds$country), decreasing = TRUE))[1]
```

```
## [1] "England"
```

```
ds[c(which(is.na(ds$country))),]$country <- "England"
ds$country = as.factor(ds$country) # convertimos a factor
```

Revisamos que hemos sustituido adecuadamente las variables `age` y `country`.

```
colSums(is.na(ds))
```

```
##     name   gender      age    class embarked  country ticketno     fare
##        0        0        0        0        0        0      891      916
##    sibsp    parch survived
##      900      900        0
```

Como vemos las variables `ticketno`, `fare`, `sibsp`, `parch` quedan por tratar. No hemos encontrado interés alguno en la variable `ticketno`.

En el caso concreto de la variable `fare`, la cual indica el precio del pasaje, se destaca que el valor es NA para miembros de la tripulación, músicos y empleados de la compañía naviera (valor numérico).

Tanto para la variable `sibsp` (número de esposas/hermanos a bordo) como para la variable `parch` (número de padres/hijos a bordo) lo que si se hará es sustituir los valores NA por el valor de la media en cada variable.

Revisamos que hemos sustituido adecuadamente las variables `sibsp` y `parch`.

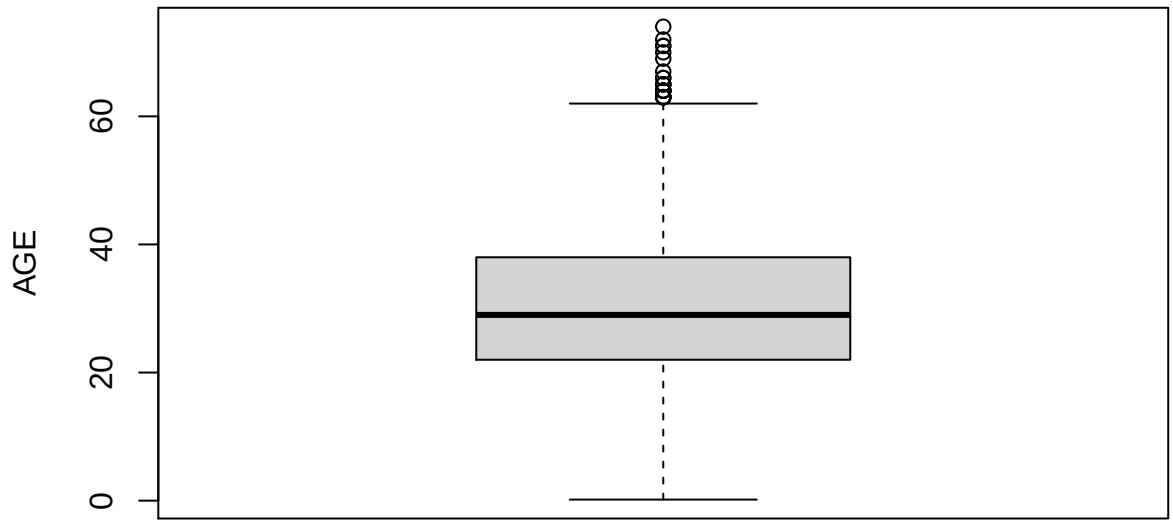
```
colSums(is.na(ds))
```

```
##     name   gender      age    class embarked  country ticketno     fare
##        0        0        0        0        0        0      891      916
##    sibsp    parch survived
##      900      900        0
```

3.2. Identificación y tratamiento de valores extremos.

Procedemos a visualizar mediante un diagrama de cajas algunas variables numéricas para detectar posibles outliers.

```
boxplot(ds$age, ylab="AGE")
```



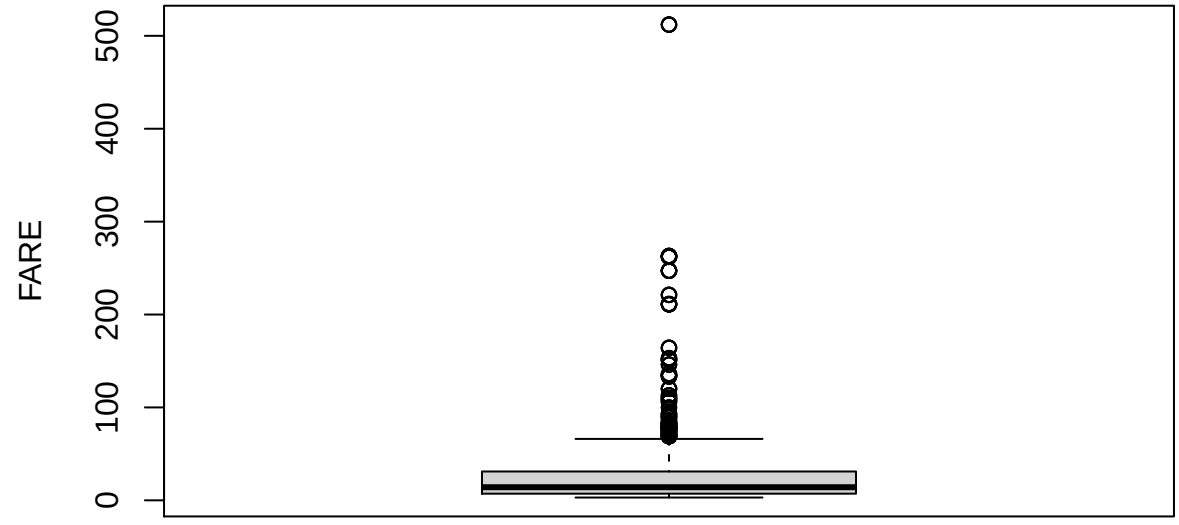
3.2.1 Age

```
sort(boxplot.stats(ds$age)$out)
```

```
##  [1] 63 63 63 63 63 63 63 64 64 64 64 64 64 65 65 65 66 66 67 69 70 71 71 72 74
```

Tras ver los datos, parecen edades con sentido. Observamos que el valor máximo es 74 y el mínimo 63 para aquellos valores que sobresalen en la distribución de la población.

```
boxplot(ds$fare, ylab="FARE")
```



3.2.2 Fare

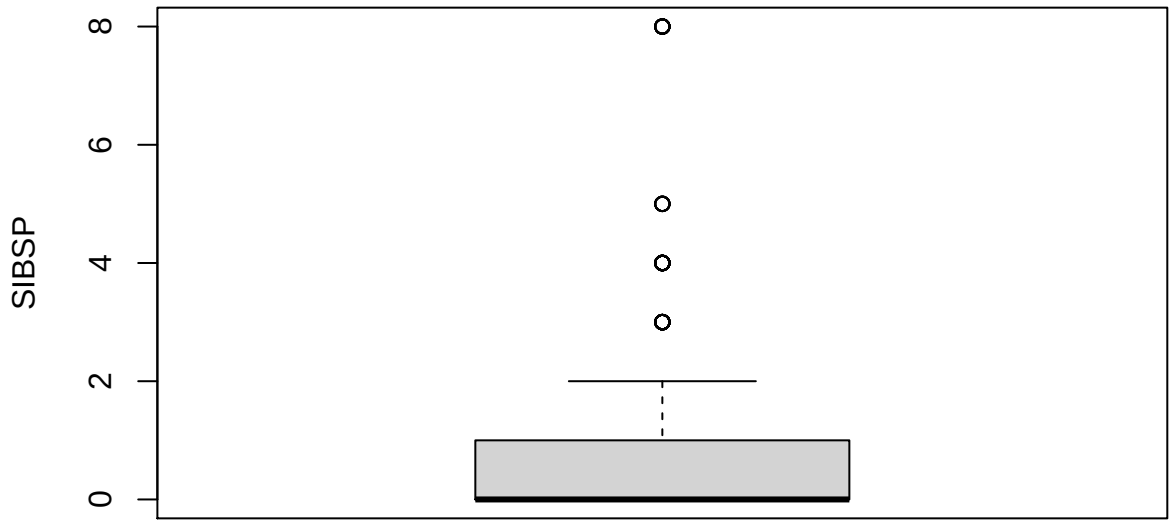
Para la variable `fare` si encontramos valores realmente extremos. Vemos que al tratarse de un primer viaje no todos los pasajeros pagaron lo mismo. De hecho el salto con respecto a la media es significativo. Estos valores extremos es interesante conservarlos pues destaca la jerarquía social de la persona a bordo y seguramente nos sea interesante de cara a saber quienes sobrevivieron.

Con esta tabla podemos ver cuantas personas pagaron cada cantidad de dinero.

```
table(sort(boxplot.stats(ds$fare)$out))
```

```
##
##    69.06    69.11       71  71.0508     73.1   75.041    75.05   76.051
##        2       11        2        2        7        2        2        2
##  76.1407  77.0509  77.1902  78.0504    78.17       79    79.04    79.13
##        3        2        3        2        3        1        4        3
##       80  81.1702  82.0305  82.0504  83.0302  83.0906     86.1  89.0201
##        3        3        2        2        6        2        3        2
##       90  91.0107     93.1      100 106.0806   108.18 110.1708 113.0506
##        3        2        4        2        3        3        4        3
##      120   133.13    134.1 135.1208 136.1507 146.1005   151.16 153.0903
##        4        2        5        3        2        3        6        3
## 164.1704 211.0609    211.1 211.6009 221.1507 247.1005 247.1006 262.0706
##        4        3        5        1        4        3        5        7
##      263 512.0607
##        6        4
```

```
boxplot(ds$sibsp, ylab="SIBSP")
```



3.2.3 Sibsp

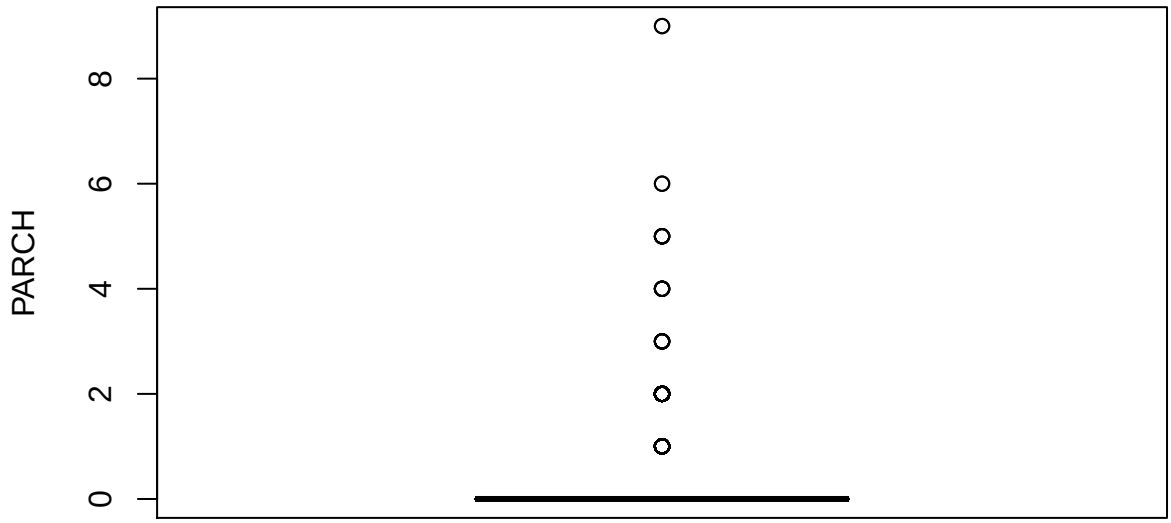
```
table(sort(boxplot.stats(ds$sibsp)$out))
```

```
##
##  3  4  5  8
## 20 22  6  9
```

Recordemos, la variable `sibsp` indica número de esposas/hermanos a bordo. Por lo tanto si encontramos el valor **3** deberíamos encontrar valores múltiplos de 4 en el dataset. Contando manualmente mi razonamiento parece cierto en los grupos de **3** hermanos, de **5** y de **8**. Sin embargo para **4** falla. Así que puede ser un error, o puede ser que fueran 3 hermanos + 1 pareja. Y cuando le preguntasen a dicha pareja respondiese con 1.

Deducimos que la mayoría de pasajeros del titanic no viajaban ni con hermanos ni con parejas.

```
boxplot(ds$parch, ylab="PARCH")
```



3.2.4 Parch

```
table(sort(boxplot.stats(ds$parch)$out))
```

```
##
##   1   2   3   4   5   6   9
## 170 113   8   6   6   2   2
```

En el caso de la variable `parch` es el número de padres/hijos a bordo. Tampoco vamos a eliminar los outliers pues parecen tener sentido. Observamos que la mayoría de pasajeros del Titanic no tenían ni padres ni hijos a bordo.

4. Análisis de los datos.

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

A continuación elegiremos qué datos utilizaremos, qué datos eliminaremos y si es necesario crear nuevas variables para llevar a cabo nuestro análisis.

4.1.1. Eliminación de variables. Descartaremos de nuestro dataset las variables **name** y **ticketno** ya que no aportan información de interés. En el caso de **ticketno**, podría dar información respecto a si el sujeto es pasajero o miembro de la tripulación, pero el mismo dato se puede obtener de la variable **fare** que sí es de interés para nuestro análisis.

```
ds = select(ds, -name, -ticketno)
str(ds)
```

```
## 'data.frame':    2207 obs. of  9 variables:
##  $ gender  : Factor w/ 2 levels "female","male": 2 2 2 1 1 2 2 1 2 2 ...
##  $ age     : num  42 13 16 39 16 25 30 28 27 20 ...
##  $ class   : Factor w/ 7 levels "1st","2nd","3rd",..: 3 3 3 3 3 3 2 2 3 3 ...
```

```
##  $ embarked: Factor w/ 4 levels "B","C","Q","S": 4 4 4 4 4 4 2 2 2 4 ...
##  $ country : Factor w/ 48 levels "Argentina","Australia",..: 45 45 45 15 31 45 17 17 27 16 ...
##  $ fare    : num  7.11 20.05 20.05 20.05 7.13 ...
##  $ sibsp   : int  0 0 1 1 0 0 1 1 0 0 ...
##  $ parch   : int  0 2 1 1 0 0 0 0 0 0 ...
##  $ survived: Factor w/ 2 levels "no","yes": 1 1 1 2 2 2 1 2 2 2 ...
```

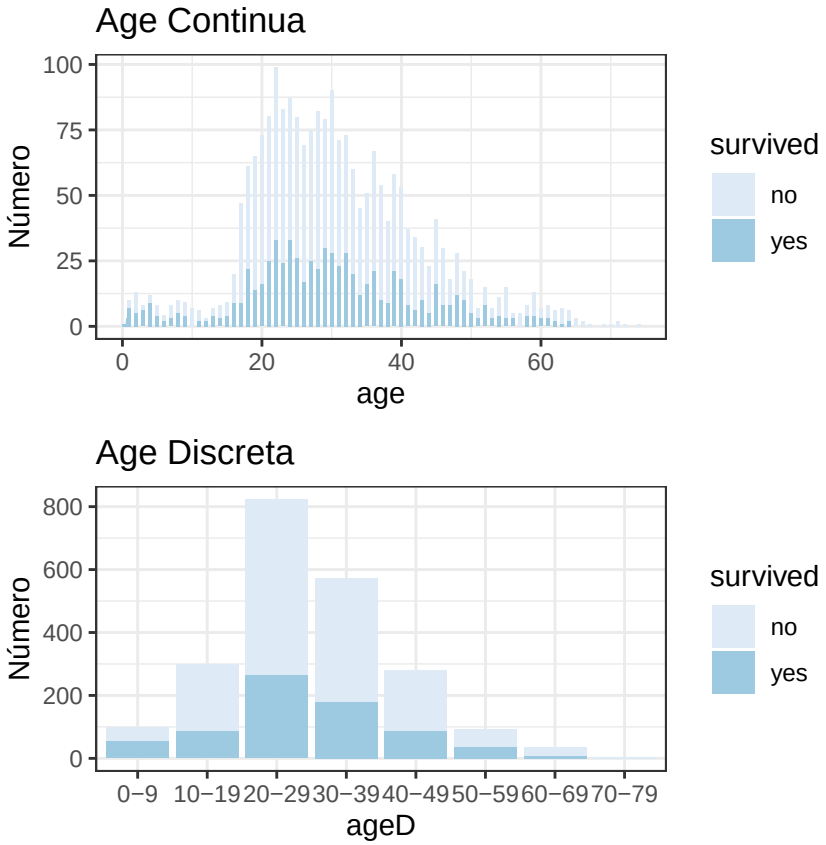
4.1.2. Generación de nuevas variables. Generaremos una nueva variable que se corresponde a la **edad discretizada**.

```
#Edad discretizada con un método simple de intervalos de igual amplitud.
summary(ds[,"age"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1667 22.0000 29.0000 30.4318 38.0000 74.0000
```

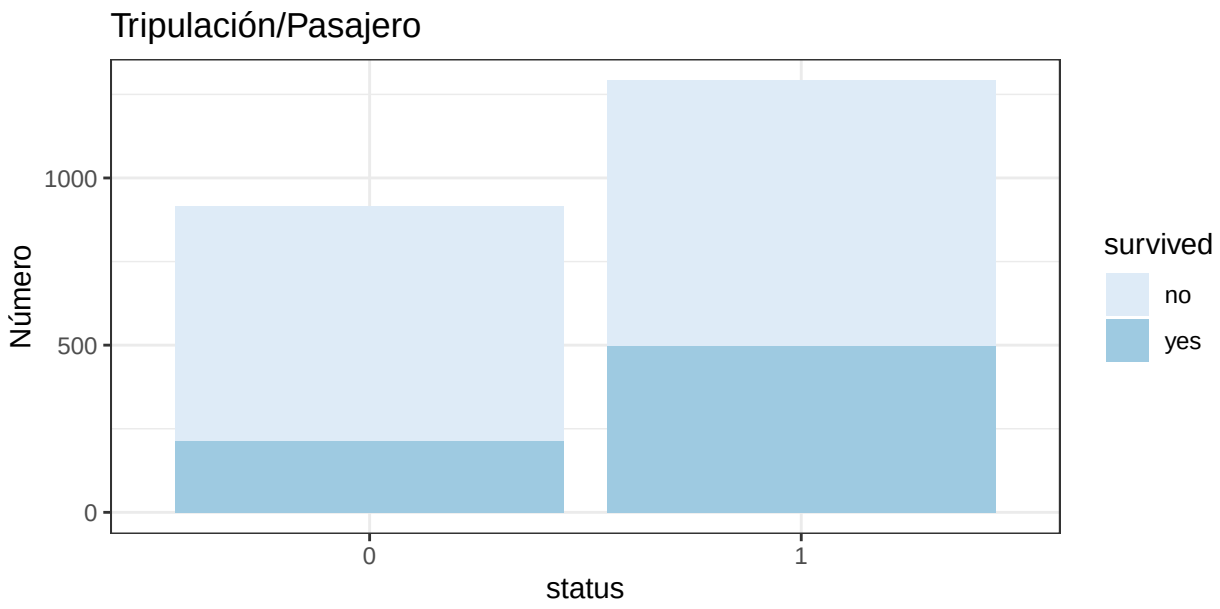
```
ds["ageD"] = cut(ds$age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c("0-9", "10-19", "20-29", "3
g1 <- ggplot(ds, aes(x = age, fill=survived))+geom_bar(width=0.5)+scale_fill_brewer(palette="Blues")+th
g2 = ggplot(ds, aes(x = ageD, fill=survived)) + geom_bar()+scale_fill_brewer(palette="Blues")+theme(asp
gridExtra::grid.arrange(g1, g2, ncol=1)
```

```
## Warning: position_stack requires non-overlapping x intervals
```



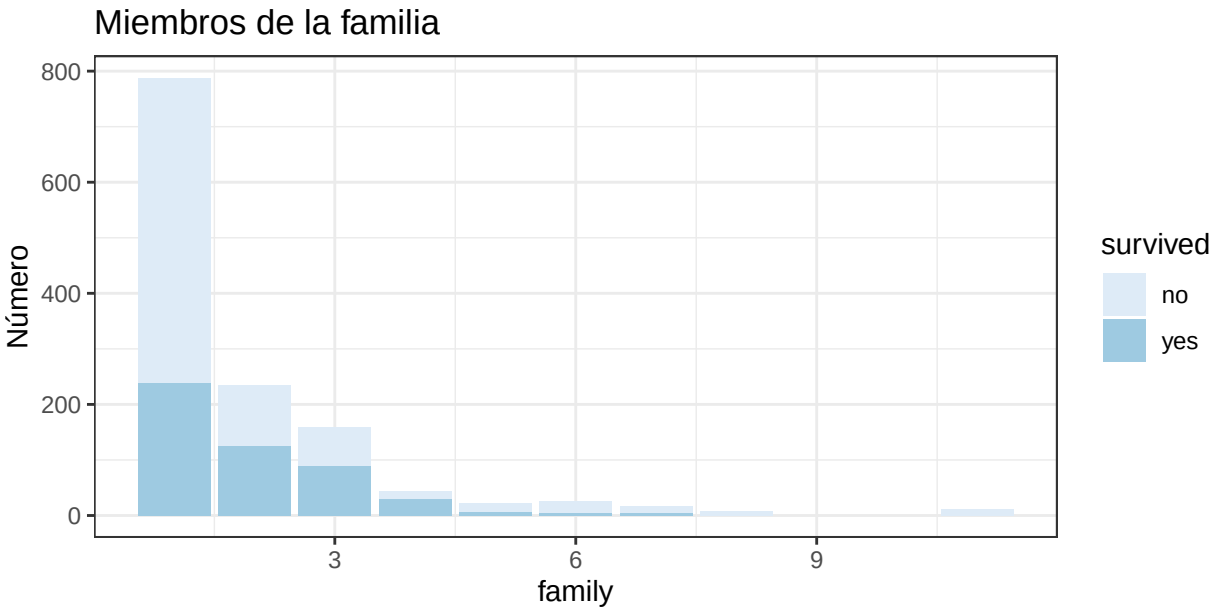
A continuación crearemos una variable que nos dirá si el sujeto es **miembro de la tripulación o pasajero**.

```r
#Nueva variable para identificar pasajeros y miembros de la tripulación
ds$status = ds$fare
ds$status[is.na(ds$status)] = 0
ds$status[ds$status != 0] = 1
ds$status = as.factor(ds$status)
g3 = ggplot(ds, aes(x = status, fill=survived))+ geom_bar()+scale_fill_brewer(palette="Blues")+theme(as
g3
```



Por último, generaremos una variable que nos indicará el tamaño de la familia de cada sujeto.

```r
#Nueva variable para calcular el tamaño de la familia entre los pasajeros
ds$sibspN = as.numeric(ds$sibsp)
ds$parchN = as.numeric(ds$parch)
ds$family = ds$sibspN + ds$parchN +1
ds = select(ds, -sibspN, -parchN)
g4 = ggplot(ds,aes(x=family,fill=survived))+geom_bar()+scale_fill_brewer(palette="Blues")+theme(aspect.
g4
```



4.1.3. Selección del grupo de datos para el análisis. Una vez tenemos el dataset con las variables que necesitamos, seleccionaremos los grupos de datos que puede ser interesante analizar:

- Pasajeros que hayan sobrevivido en función de la edad: **survived** y **age**/**ageD**
- Pasajeros que hayan sobrevivido en función del genero: **survived** y **genre**
- Pasajeros que hayan sobrevivido en función del puerto de embarque: **survived** y **embarked**
- Pasajeros que hayan sobrevivido en función del país de origen: **survived** y **country**
- Pasajeros que hayan sobrevivido en función de la clase: **survived** y **class**
- Pasajeros que hayan sobrevivido en función del número de familiares: **survived** y **sibsp**
- Pasajeros que hayan sobrevivido en función del número de familiares: **survived** y **parch**
- Pasajeros que hayan sobrevivido en función del tamaño de la familia con la que viajaban: **survived** y **family**
- Categoría del billete en función del puerto de embarque: **class** y **embarked**
- Categoría del billete en función del país de origen: **class** y **country** Con los datos anteriores, al llevar a cabo el análisis se verá si resulta de interés hacer alguna combinación de las variables anteriores

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

A la hora de realizar un análisis de normalidad y homgeneidad de la varianza, empezaremos con un análisis descriptivo de los datos:

```
summary(ds)
```

```
##     gender          age                        class     embarked
##  female: 489   Min.   : 0.1667   1st             :324   B: 197
##  male  :1718   1st Qu.:22.0000   2nd             :284   C: 271
##                Median :29.0000   3rd             :709   Q: 123
##                Mean   :30.4318   deck crew       : 66   S:1616
##                3rd Qu.:38.0000   engineering crew:324
##                Max.   :74.0000   restaurant staff: 69
##                                  victualling crew:431
##          country          fare             sibsp           parch
```

```
##  England      :1206   Min.   :  3.030   Min.   :0.0000   Min.   :0.0000
##  United States: 264   1st Qu.:  7.181   1st Qu.:0.0000   1st Qu.:0.0000
##  Ireland      : 137   Median : 14.090   Median :0.0000   Median :0.0000
##  Sweden       : 105   Mean   : 33.405   Mean   :0.4996   Mean   :0.3856
##  Lebanon      :  71   3rd Qu.: 31.061   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Finland      :  54   Max.   :512.061   Max.   :8.0000   Max.   :9.0000
##  (Other)      : 370   NA's   :916       NA's   :900      NA's   :900
##  survived        ageD     status       family
##  no :1496   20-29  :824   0: 916   Min.   : 1.000
##  yes: 711   30-39  :572   1:1291   1st Qu.: 1.000
##             10-19  :299            Median : 1.000
##             40-49  :280            Mean   : 1.885
##             0-9    :100            3rd Qu.: 2.000
##             50-59  : 93            Max.   :11.000
##             (Other): 39            NA's   :900
```

A partir de los datos obtenidos, podemos calcular la normalidad de los tres atributos numéricos: **age**, **fare**, y **family**. Para ello utilizaremos el test de Saphiro-Wilk:

```
shapiro.test(ds$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ds$age
## W = 0.98054, p-value < 2.2e-16
```

```
shapiro.test(ds$fare)
```

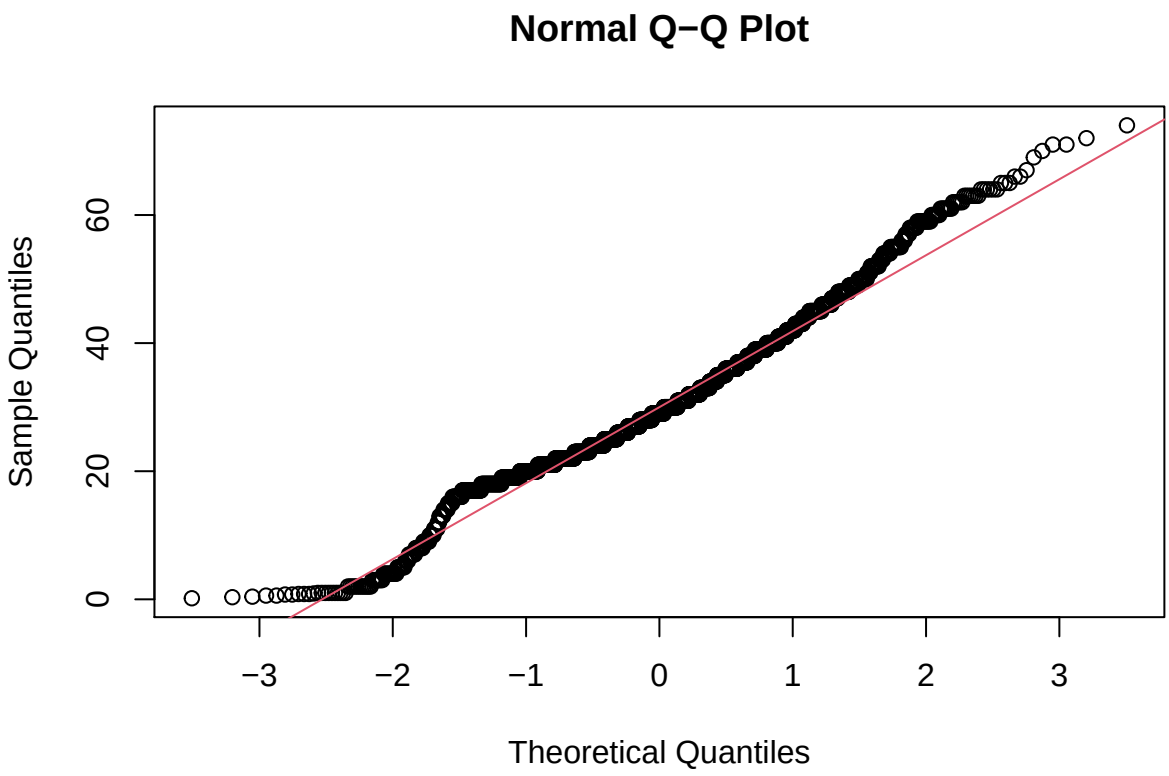
```
##
##  Shapiro-Wilk normality test
##
## data:  ds$fare
## W = 0.52374, p-value < 2.2e-16
```

```
shapiro.test(ds$family)
```

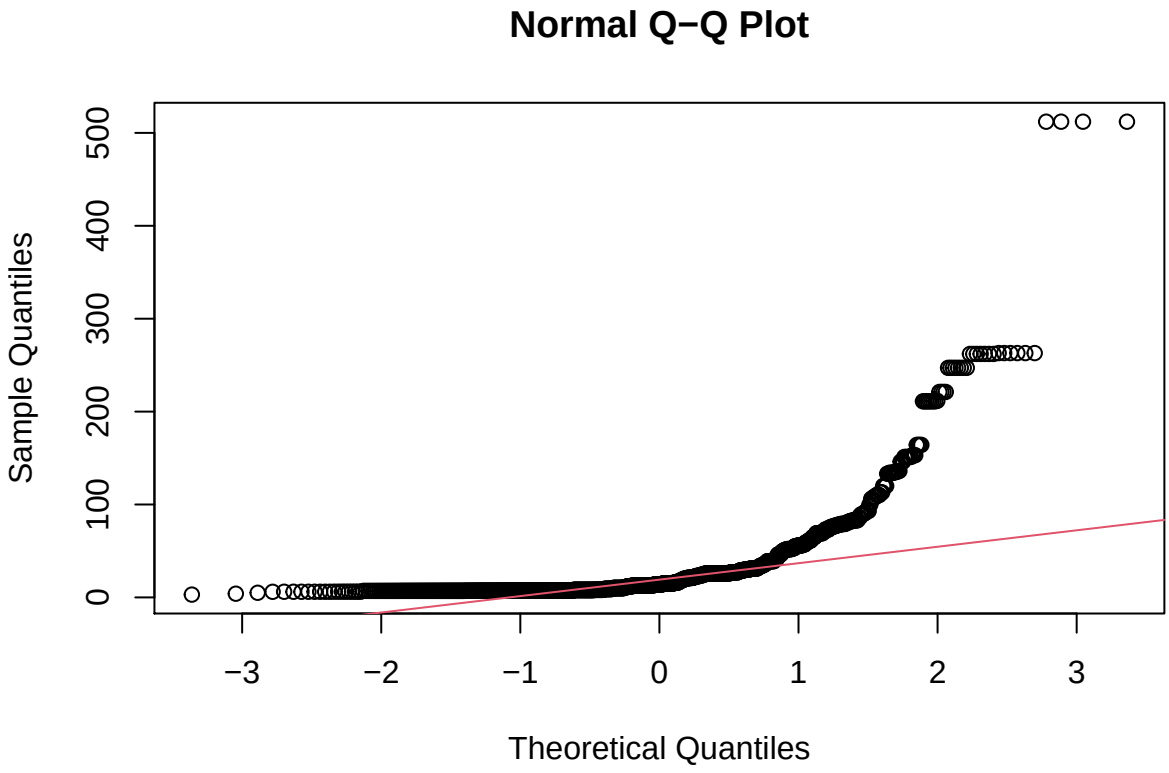
```
##
##  Shapiro-Wilk normality test
##
## data:  ds$family
## W = 0.60956, p-value < 2.2e-16
```

A continuación se analiza el valor de p. En el caso de que sea mayor que el nivel de significancia, se acepta la hipótesis nula y se concluye que la variable tiene una distribución normal. En nuestro caso todas las variables analizadas presentan valores inferiores al nivel de signficancia, por lo que podemos concluir con un 95% de confianza que no presentan una distribución normal. Veámoslo a continuación de forma gráfica:

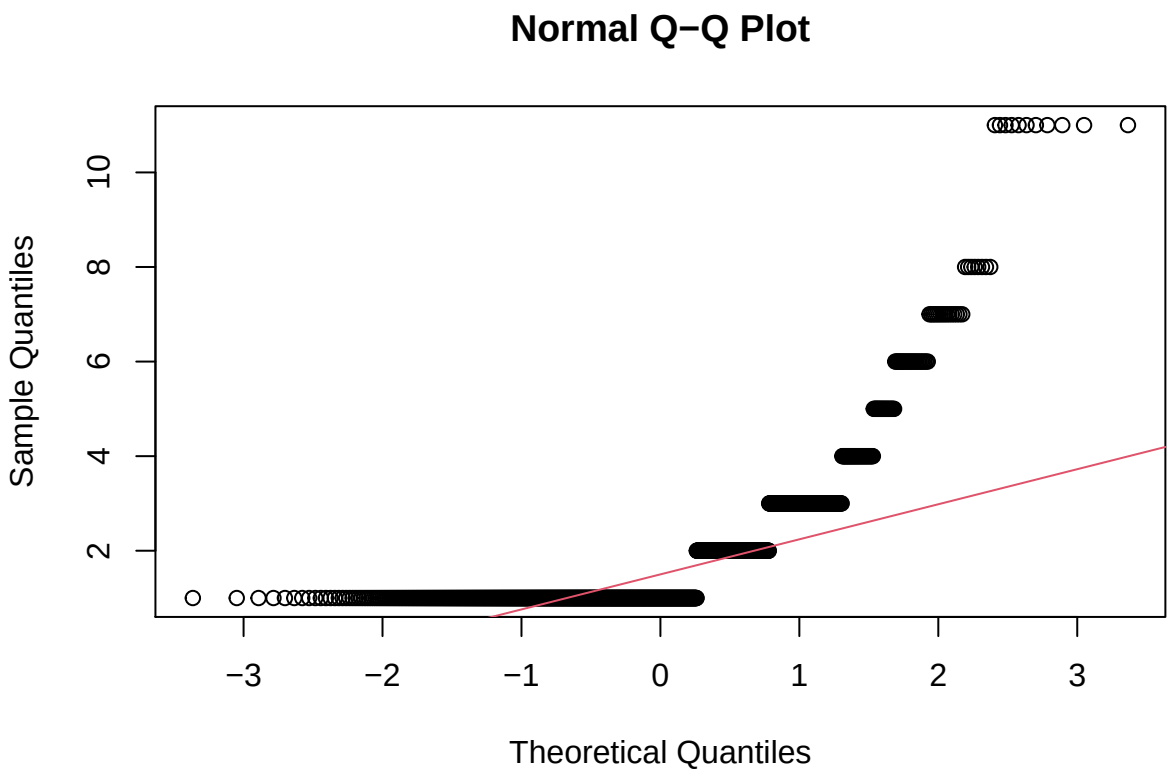
```
qqnorm(ds$age);qqline(ds$age, col = 2)
```




```
qqnorm(ds$fare);qqline(ds$fare, col = 2)
```



```
qqnorm(ds$family);qqline(ds$family, col = 2)
```



Seguidamente, pasamos a estudiar la homogeneidad de varianzas mediante la aplicación de un test de Fligner-Killeen. En este caso, estudiaremos la homogeneidad de las cuatro variables anteriormente mencionadas respecto a la variable survived. En nuestro test, la hipótesis nula consiste en que ambas varianzas son iguales.

```
fligner.test(age ~ survived, data = ds)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  age by survived
## Fligner-Killeen:med chi-squared = 3.2538, df = 1, p-value = 0.07126
```

```
fligner.test(fare ~ survived, data = ds)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  fare by survived
## Fligner-Killeen:med chi-squared = 138.14, df = 1, p-value < 2.2e-16
```

```
fligner.test(family ~ survived, data = ds)
```

```
##
```

```
##  Fligner-Killeen test of homogeneity of variances
##
## data:  family by survived
## Fligner-Killeen:med chi-squared = 31.783, df = 1, p-value = 1.724e-08
```

Una vez ejecutados los test, podemos aceptar la hipótesis nula en aquellos casos en los que el valor de p sea mayor que el valor de significancia (0,05). Por lo tanto, para la variable **age**, con p-valor>0,05 diremos con un 95% de confianza que su varianza es la misma tanto para los supervivientes como para los que no. Siguiendo el mismo razonamiento, para **fare** y **family**, con p-valor<=0,05, afirmaremos que las varianzas de estas variables son diferentes para los dos grupos de la variable "survived".

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

Nos interesa realizar un estudio de cómo se relacionan las variables para determinar si una persona sobrevive o no al hundimiento del Titanic. Para ello se hará uso de la variable dicotómica `survived`. Sería interesante analizar si existe asociación entre la variable dependiente `survived` y las variables explicativas que queramos usar de cara a la construcción del modelo predictivo de regresión logística.

```
chisq.test(table(ds$survived, ds$gender))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(ds$survived, ds$gender)
## X-squared = 485.87, df = 1, p-value < 2.2e-16
```

```
chisq.test(table(ds$survived, ds$age))
```

```
## Warning in chisq.test(table(ds$survived, ds$age)): Chi-squared approximation may
## be incorrect

##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$age)
## X-squared = 110.3, df = 78, p-value = 0.009436
```

```
chisq.test(table(ds$survived, ds$class))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$class)
## X-squared = 252.24, df = 6, p-value < 2.2e-16
```

```
chisq.test(table(ds$survived, ds$embarked))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$embarked)
## X-squared = 90.351, df = 3, p-value < 2.2e-16
```

```
chisq.test(table(ds$survived, ds$country))
```

```
## Warning in chisq.test(table(ds$survived, ds$country)): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$country)
## X-squared = 189.4, df = 47, p-value < 2.2e-16
```

```
chisq.test(table(ds$survived, ds$fare))
```

```
## Warning in chisq.test(table(ds$survived, ds$fare)): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$fare)
## X-squared = 556.14, df = 275, p-value < 2.2e-16
```

```
chisq.test(table(ds$survived, ds$sibsp))
```

```
## Warning in chisq.test(table(ds$survived, ds$sibsp)): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$sibsp)
## X-squared = 43.257, df = 6, p-value = 1.037e-07
```

```
chisq.test(table(ds$survived, ds$parch))
```

```
## Warning in chisq.test(table(ds$survived, ds$parch)): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$parch)
## X-squared = 53.565, df = 7, p-value = 2.867e-09
```

```
chisq.test(table(ds$survived, ds$family))
```

```
## Warning in chisq.test(table(ds$survived, ds$family)): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(ds$survived, ds$family)
## X-squared = 102.78, df = 8, p-value < 2.2e-16
```

Como podemos observar las variables `gender`, `class`, `embarked` afectan a la supervivencia tras el hundimiento del Titanic. Las variables `age`, `sibsp`, `parch` y `family` no parecen afectar a la supervivencia. Además en muchos tests parece que la aproximación parece no ser del todo correcta.

Ahora bien, de estas variables que hemos detectado que pueden afectar a la supervivencia, vamos a analizar cómo se comportan a la hora de construir el modelo.

```
model_gender <- glm(formula = ds$survived~factor(ds$gender), data = ds, family = binomial)
summary(model_gender)
```

```
##
## Call:
## glm(formula = ds$survived ~ factor(ds$gender), family = binomial,
##     data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6278  -0.6772  -0.6772   0.7862   1.7806
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.0158     0.1024   9.924   <2e-16 ***
## factor(ds$gender)male  -2.3718     0.1185 -20.009   <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: 2774.1  on 2206  degrees of freedom
## Residual deviance: 2308.8  on 2205  degrees of freedom
## AIC: 2312.8
##
## Number of Fisher Scoring iterations: 4
```

Vamos a calcular los OR para saber si su efecto sobre el modelo es un factor protector o no.

```
exp(cbind(coef(model_gender), confint(model_gender)))
```

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

```
##                                       2.5 %    97.5 %
## (Intercept)           2.76153846 2.26626223 3.3863167
## factor(ds$gender)male 0.09331272 0.07375952 0.1174152
```

Por ejemplo, y curiosamente, en este caso parece ser que ser hombre tiene un factor protector en el modelo.

Vamos a construir el modelo en función de todas las variable que consideramos afectan al modelo.

```
model_gender_class_embarked <- glm(formula = ds$survived~factor(ds$gender)+factor(ds$class)+factor(ds$em
summary(model_gender_class_embarked)
```

```
##
## Call:
## glm(formula = ds$survived ~ factor(ds$gender) + factor(ds$class) +
##     factor(ds$embarked), family = binomial, data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3067  -0.6722  -0.4598   0.7183   2.5893
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       1.6285     0.2825   5.765 8.18e-09 ***
## factor(ds$gender)male            -2.5958     0.1447 -17.933  < 2e-16 ***
## factor(ds$class)2nd              -0.7487     0.2074  -3.610 0.000306 ***
## factor(ds$class)3rd              -1.6130     0.1827  -8.831  < 2e-16 ***
## factor(ds$class)deck crew         1.4102     0.3119   4.521 6.16e-06 ***
## factor(ds$class)engineering crew -0.5529     0.2086  -2.651 0.008025 **
## factor(ds$class)restaurant staff -2.6339     0.6274  -4.198 2.69e-05 ***
## factor(ds$class)victualling crew -0.6899     0.2045  -3.374 0.000740 ***
## factor(ds$embarked)C              0.9594     0.2767   3.467 0.000526 ***
## factor(ds$embarked)Q              0.3864     0.3266   1.183 0.236761
## factor(ds$embarked)S              0.2846     0.2097   1.358 0.174600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2774.1  on 2206  degrees of freedom
## Residual deviance: 2098.0  on 2196  degrees of freedom
## AIC: 2120
##
## Number of Fisher Scoring iterations: 5
```

Se ve claramente cómo el género, la clase y el puerto de embarque afectan al modelo notablemente. Dejamos como pendiente crear un modelo con alguna variable descartada. Para comprobar que la decisión tomada fuera la correcta.

No obstante, primero vamos a calcular los OR para todas las variables seleccionadas con el objetivo de saber si su efecto sobre el modelo es un factor protector o no.

```
exp(cbind(coef(model_gender_class_embarked), confint(model_gender_class_embarked)))
```

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

```
##                                                 2.5 %     97.5 %
## (Intercept)                          5.09633236 2.91974411 8.84795951
## factor(ds$gender)male                0.07458670 0.05590166 0.09863276
## factor(ds$class)2nd                  0.47296573 0.31428726 0.70906925
## factor(ds$class)3rd                  0.19928724 0.13885631 0.28427891
```

```
## factor(ds$class)deck crew        4.09689021 2.24292442 7.64588768
## factor(ds$class)engineering crew 0.57527269 0.38163616 0.86502808
## factor(ds$class)restaurant staff 0.07179801 0.01664504 0.21089577
## factor(ds$class)victualling crew 0.50163365 0.33571984 0.74868484
## factor(ds$embarked)C             2.61023028 1.52499576 4.51696074
## factor(ds$embarked)Q             1.47173377 0.77664692 2.79788891
## factor(ds$embarked)S             1.32927079 0.88905584 2.02597687
```

Se destaca como factor dañino el valor **deck crew** para la variable `class`. Es decir, la tripulación de cubierta fue un factor dañino en la supervivencia en el Titanic. Le continúan como factores dañinos en la variable `embarked` los valores **C** (Cherbourg), **Q** (Queenstown), **S** (Southampton) en dicho orden.

Por curiosidad vamos a crear un nuevo modelo con una de las variables anteriormente descartadas para observar su comportamiento.

```
model_fail <- glm(formula = ds$survived~factor(ds$gender)+factor(ds$class)+factor(ds$embarked)+factor(ds
summary(model_fail)
```

```
##
## Call:
## glm(formula = ds$survived ~ factor(ds$gender) + factor(ds$class) +
##     factor(ds$embarked) + factor(ds$sibsp) + factor(ds$parch),
##     family = binomial, data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2178  -0.6404  -0.4614   0.6023   2.5606
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -13.5322   785.5331  -0.017 0.986256
## factor(ds$gender)male    -2.5348     0.1594 -15.903  < 2e-16 ***
## factor(ds$class)2nd      -0.8116     0.2128  -3.814 0.000137 ***
## factor(ds$class)3rd      -1.5180     0.1856  -8.179 2.85e-16 ***
## factor(ds$embarked)C     15.9023   785.5331   0.020 0.983849
## factor(ds$embarked)Q     15.4729   785.5332   0.020 0.984285
## factor(ds$embarked)S     15.3985   785.5331   0.020 0.984360
## factor(ds$sibsp)1        -0.1390     0.1806  -0.770 0.441507
## factor(ds$sibsp)2        -0.1093     0.4212  -0.260 0.795218
## factor(ds$sibsp)3        -1.5840     0.5993  -2.643 0.008213 **
## factor(ds$sibsp)4        -1.6891     0.7197  -2.347 0.018939 *
## factor(ds$sibsp)5       -16.2745   833.1430  -0.020 0.984415
## factor(ds$sibsp)8       -16.4835   671.4771  -0.025 0.980415
## factor(ds$parch)1         0.8870     0.2340   3.791 0.000150 ***
## factor(ds$parch)2         0.6356     0.2910   2.184 0.028964 *
## factor(ds$parch)3         0.1065     0.8487   0.125 0.900168
## factor(ds$parch)4        -1.9536     1.1880  -1.644 0.100079
## factor(ds$parch)5        -1.4584     1.1495  -1.269 0.204568
## factor(ds$parch)6       -15.8096  1425.1340  -0.011 0.991149
## factor(ds$parch)9       -15.8096  1425.1340  -0.011 0.991149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1739.1  on 1306  degrees of freedom
## Residual deviance: 1190.8  on 1287  degrees of freedom
##   (900 observations deleted due to missingness)
## AIC: 1230.8
##
## Number of Fisher Scoring iterations: 15
```

Vemos cómo ha cambiado por completo el comportamiento del modelo. La variable `embarked` ha perdido su significancia y ya no influye en el modelo de la forma que anteriormente hacía. Además las nuevas variables agregadas `sibsp` y `parch` parece que no afectan prácticamente en el modelo.

```
model_fail_family <- glm(formula = ds$survived~factor(ds$gender)+factor(ds$class)+factor(ds$embarked)+fa
summary(model_fail_family)
```

```
##
## Call:
## glm(formula = ds$survived ~ factor(ds$gender) + factor(ds$class) +
##     factor(ds$embarked) + factor(ds$family), family = binomial,
##     data = ds)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3550  -0.6338  -0.4576   0.5973   2.6663
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -13.49276  784.42835  -0.017  0.98628
## factor(ds$gender)male  -2.55713    0.16017 -15.965  < 2e-16 ***
## factor(ds$class)2nd    -0.84225    0.21324  -3.950 7.82e-05 ***
## factor(ds$class)3rd    -1.54298    0.18626  -8.284  < 2e-16 ***
## factor(ds$embarked)C   15.93146  784.42834   0.020  0.98380
## factor(ds$embarked)Q   15.42464  784.42837   0.020  0.98431
## factor(ds$embarked)S   15.38904  784.42833   0.020  0.98435
## factor(ds$family)2      0.03671    0.19641   0.187  0.85172
## factor(ds$family)3      0.62770    0.21923   2.863  0.00419 **
## factor(ds$family)4      0.81212    0.41592   1.953  0.05087 .
## factor(ds$family)5     -1.32182    0.57035  -2.318  0.02047 *
## factor(ds$family)6     -1.31979    0.61764  -2.137  0.03261 *
## factor(ds$family)7     -0.73633    0.65163  -1.130  0.25848
## factor(ds$family)8    -15.71807  715.02726  -0.022  0.98246
## factor(ds$family)11   -15.86527  605.94303  -0.026  0.97911
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1739.1  on 1306  degrees of freedom
## Residual deviance: 1194.5  on 1292  degrees of freedom
##   (900 observations deleted due to missingness)
## AIC: 1224.5
##
## Number of Fisher Scoring iterations: 15
```

Observamos que si introducimos la variable `family` en el modelo, realmente pierde eficacia el modelo y vemos un comportamiento similar al obtenido cuando agregamos las variables `parch` y `sibsp`. Podemos realizar el test de Hosman-Lemeshow para ver la bondad de ajuste.

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```
hoslem.test(ds$survived, fitted(model_gender_class_embarked))
```

```
## Warning in Ops.factor(1, y): '-' not meaningful for factors
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  ds$survived, fitted(model_gender_class_embarked)
## X-squared = 2207, df = 8, p-value < 2.2e-16
```

Si nos fijamos en el p-value se acepta la hipótesis nula, por lo tanto, el modelo se ajusta adecuadamente.

También podemos dibujar la curva **ROC** del modelo.

```
# ROC se encuentra en el paquete pROC
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

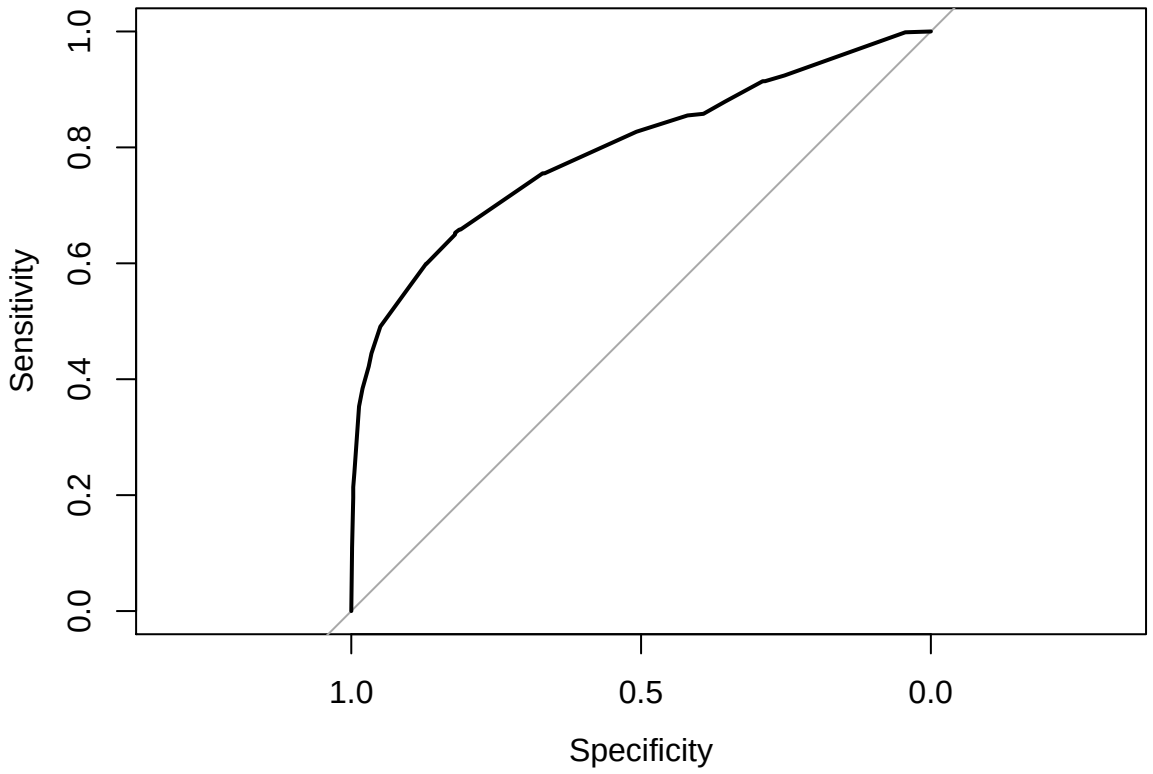
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
r_c=roc(ds$survived, predict(model_gender_class_embarked, ds, type="response") , data=ds)
```

```
## Setting levels: control = no, case = yes
```

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

```
plot(r_c)
```



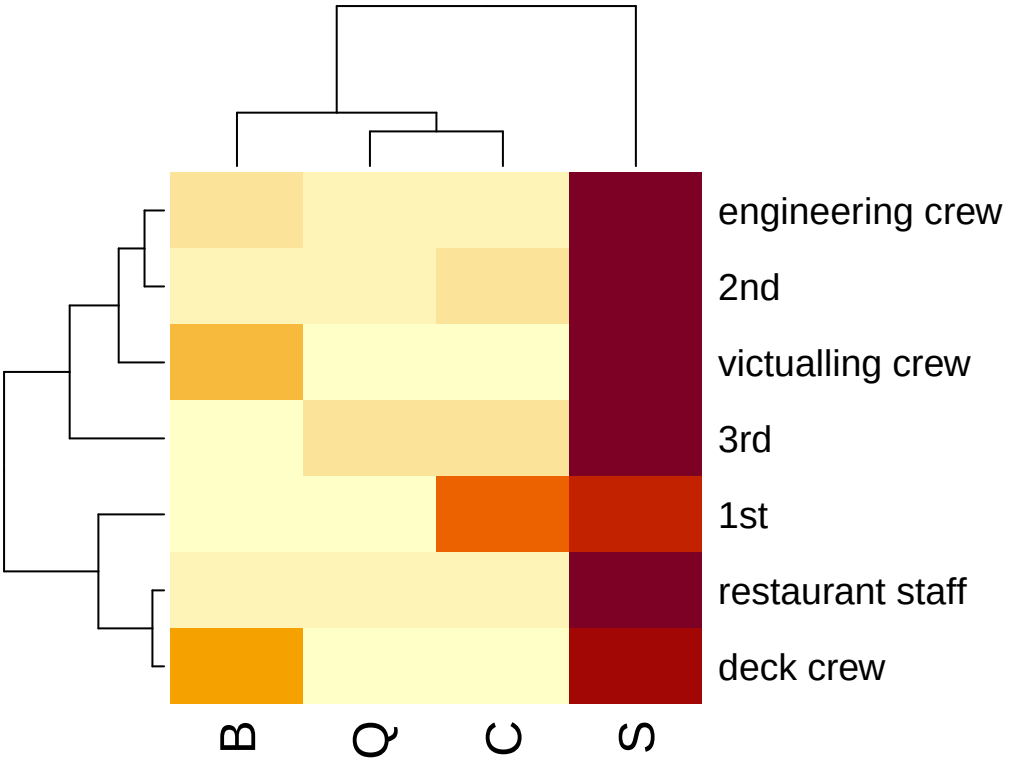

```
auc(r_c)
```

```
## Area under the curve: 0.7947
```

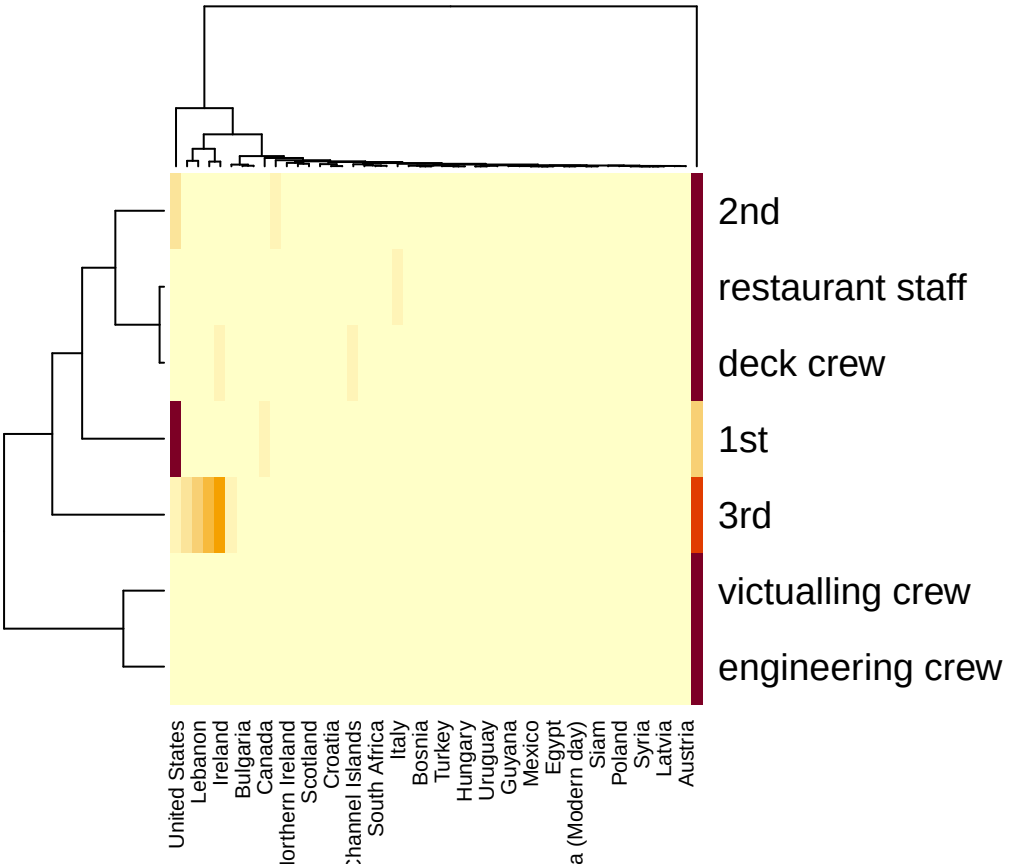
Vemos que el área debajo de la curva del modelo `model_gender_class_embarked` adquiere el valor de 0.7947 por lo tanto la capacidad del modelo para predecir supervivencia es bastante buena. Casi un 80%.

También queremos analizar la correlación existente entre la variables **class** y **embarked** y **class** y **country**. Para ello haremos uso de un mapa de calor.

```
data_cor_class_embarked <- as.matrix(table(ds$class, ds$embarked))
data_cor_class_country <- as.matrix(table(ds$class, ds$country))
heatmap(data_cor_class_embarked)
```




```
heatmap(data_cor_class_country)
```



Parece que existe

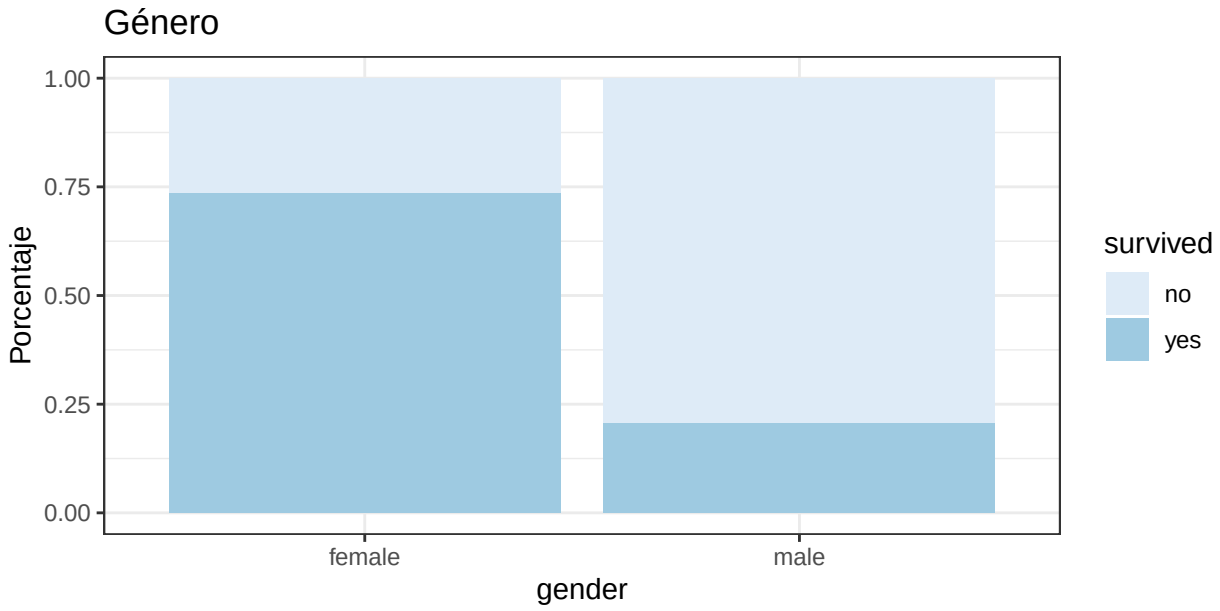
una correlación entre el puerto de embarque **C** y ser de primera clase **1st**. Luego también se ve claramente una correlación entre el puerto de embarque en **B** y formar parte de la tripulación **victualling** y **deck**. Podemos decir que la correlación está siendo sesgada por el número de pasajeros que inician el viaje en **S**.

En el caso del análisis con respecto al pais de origen se destaca el hecho de formar parte de primera clase siendo estadounidense.

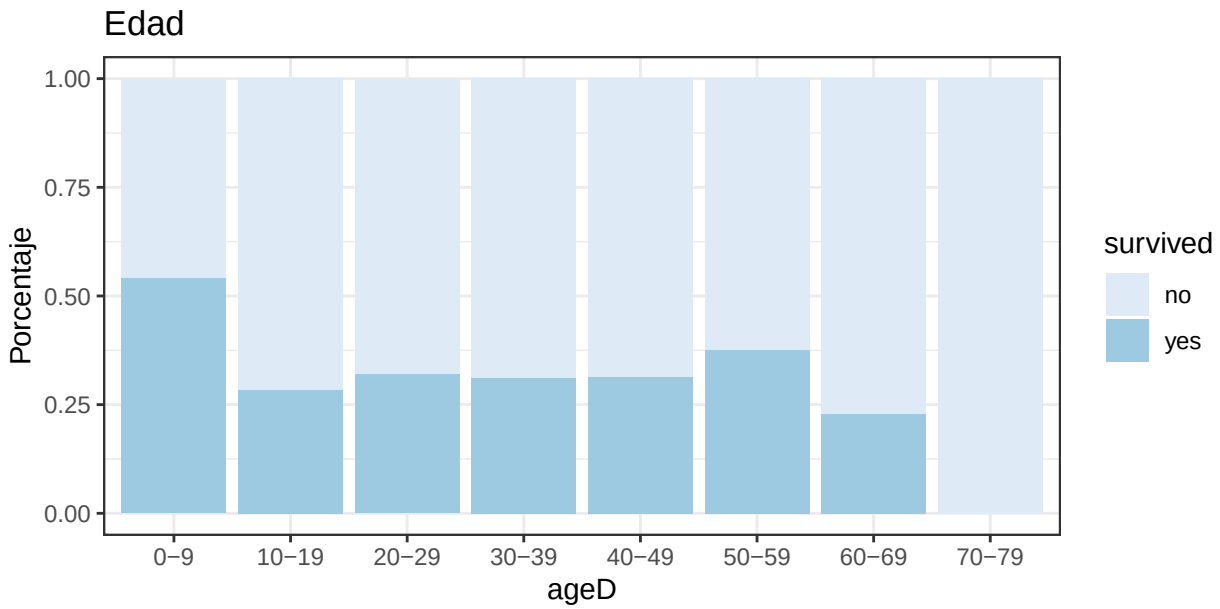
5. Representación de los resultados a partir de tablas y gráficas.

A continuación mostramos todas las gráficas de aquellas variables que hemos analizado respecto a la supervicencia en el hundimiento del titanic:

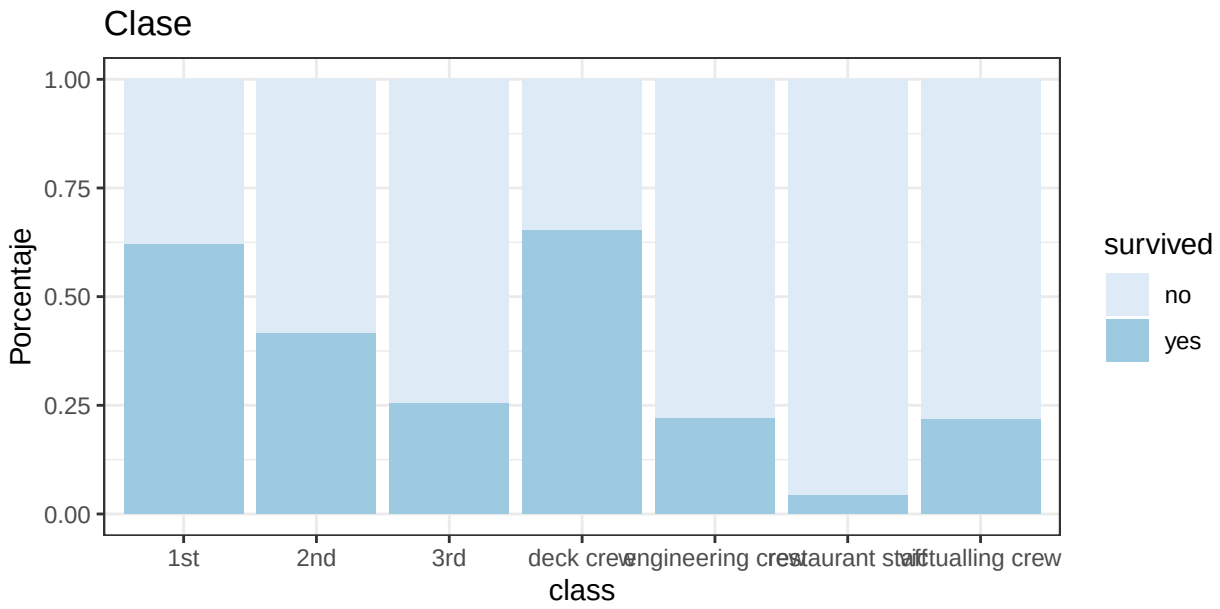
```
ggplot(ds,aes(x=gender,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+tl
```



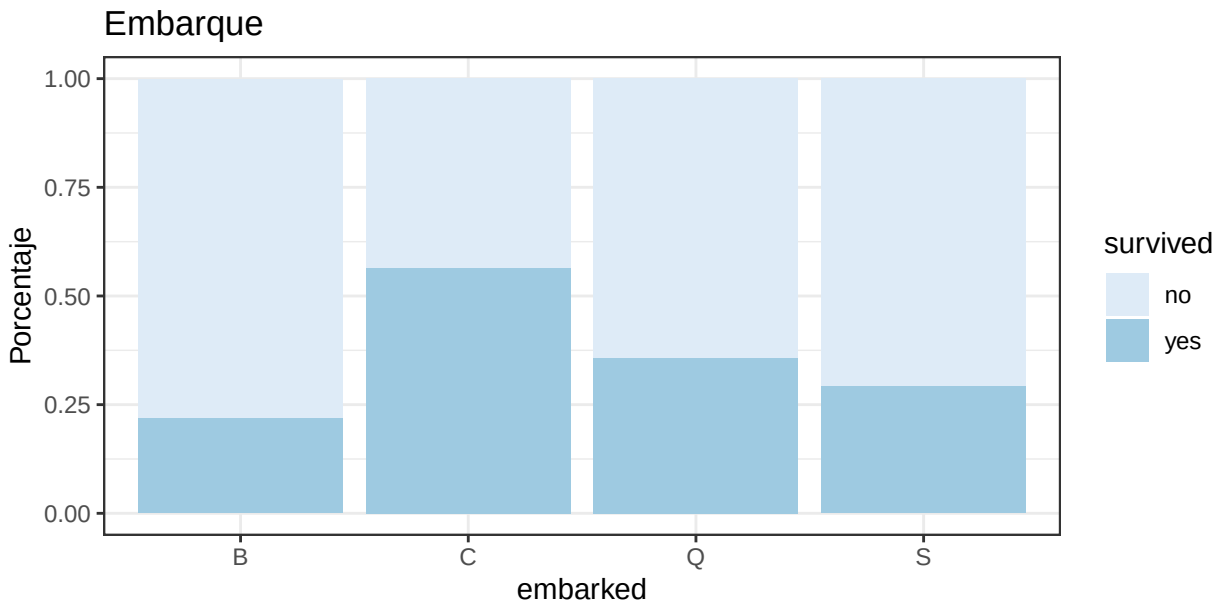

```
ggplot(ds,aes(x=ageD,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+ther
```



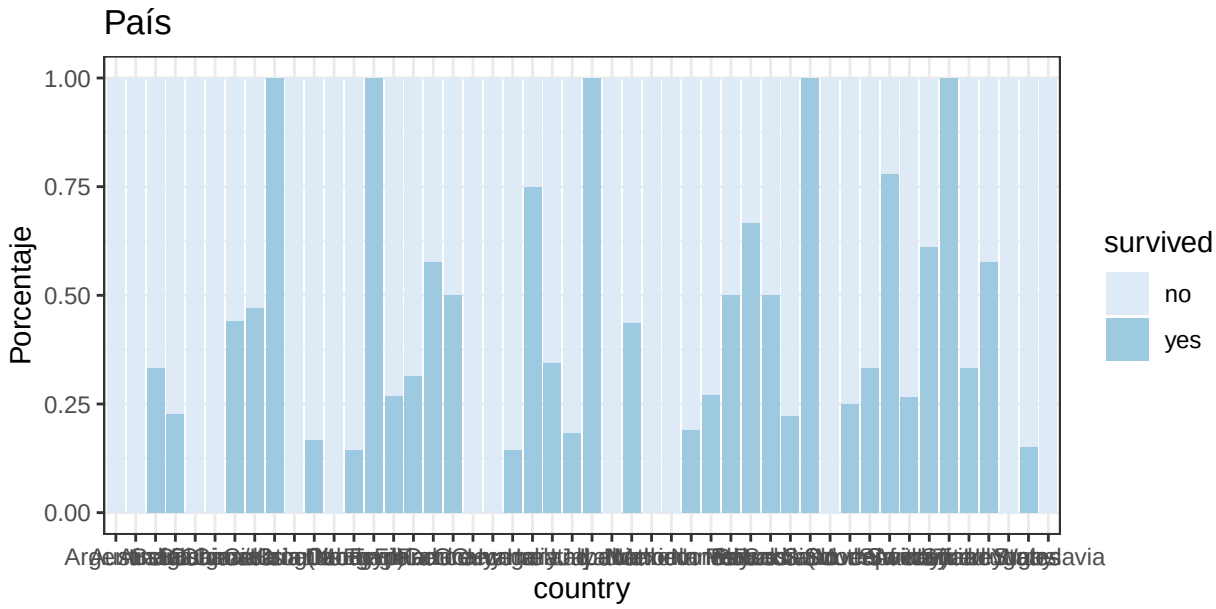

```
ggplot(ds,aes(x=class,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+th
```




```
ggplot(ds,aes(x=embarked,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+
```

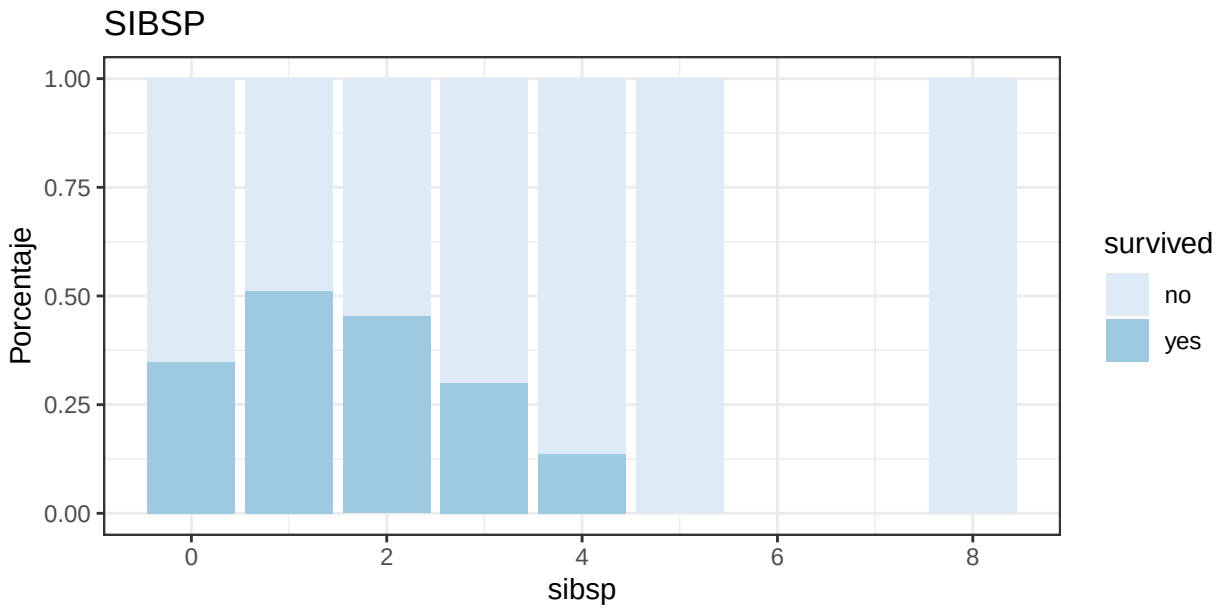



```
ggplot(ds,aes(x=country,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+
```



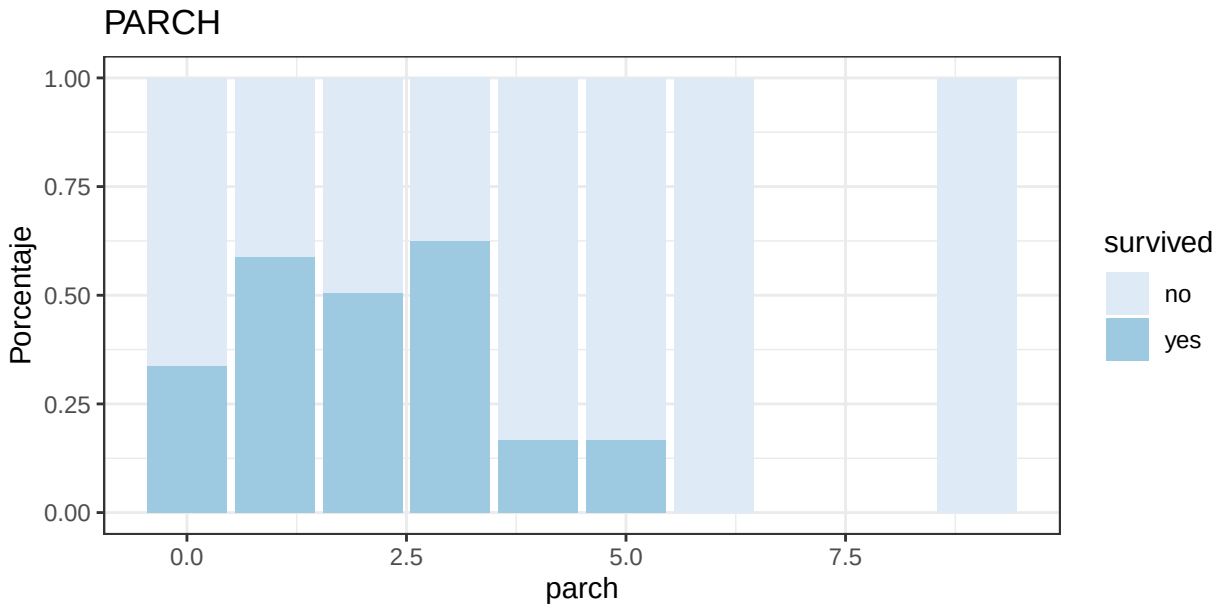

```
ggplot(ds,aes(x=sibsp,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+th
```

```
## Warning: Removed 900 rows containing non-finite values (stat_count).
```



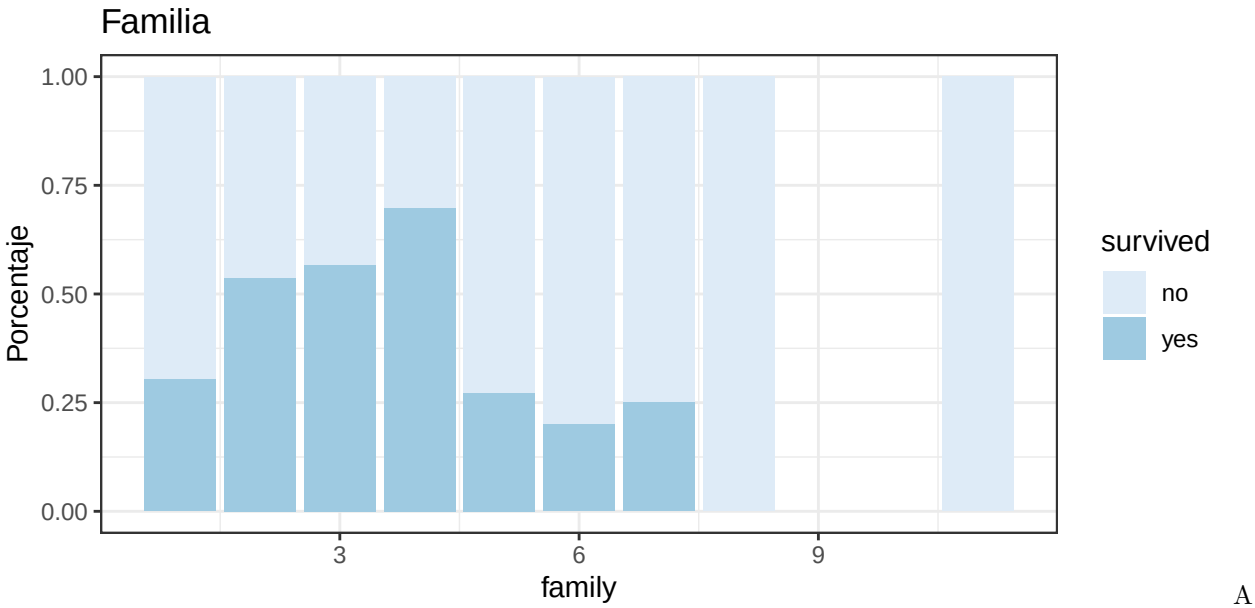

```
ggplot(ds,aes(x=parch,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+th
```

```
## Warning: Removed 900 rows containing non-finite values (stat_count).
```



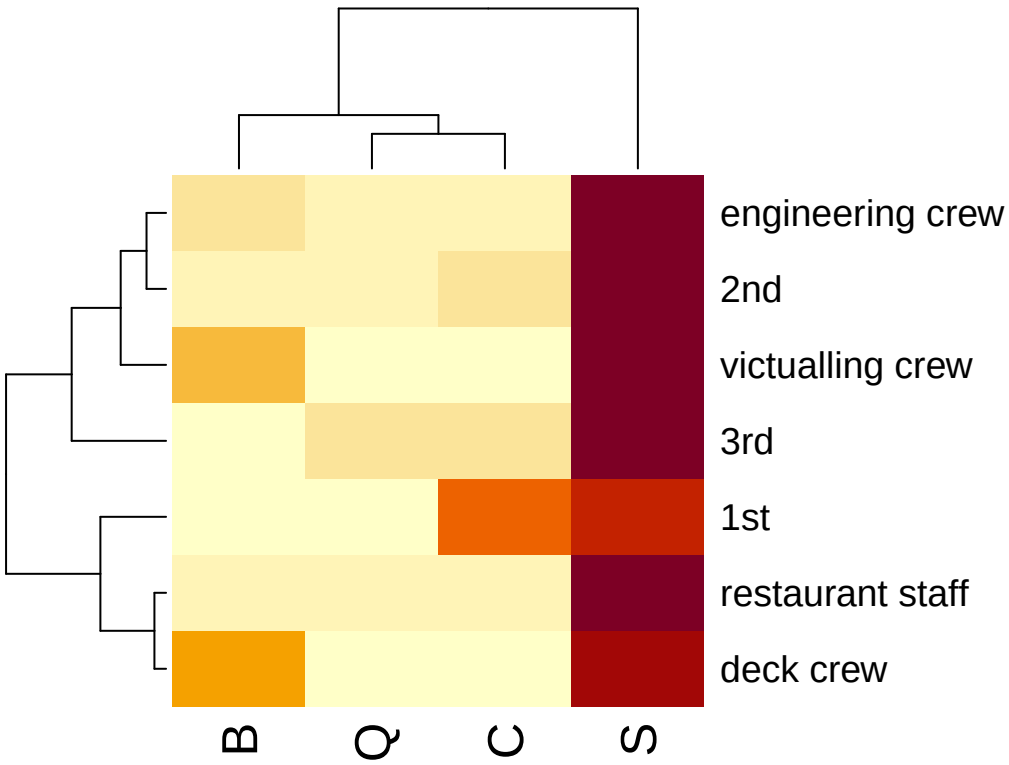

```
ggplot(ds,aes(x=family,fill=survived))+geom_bar(position = "fill")+scale_fill_brewer(palette="Blues")+t
```

```
## Warning: Removed 900 rows containing non-finite values (stat_count).
```

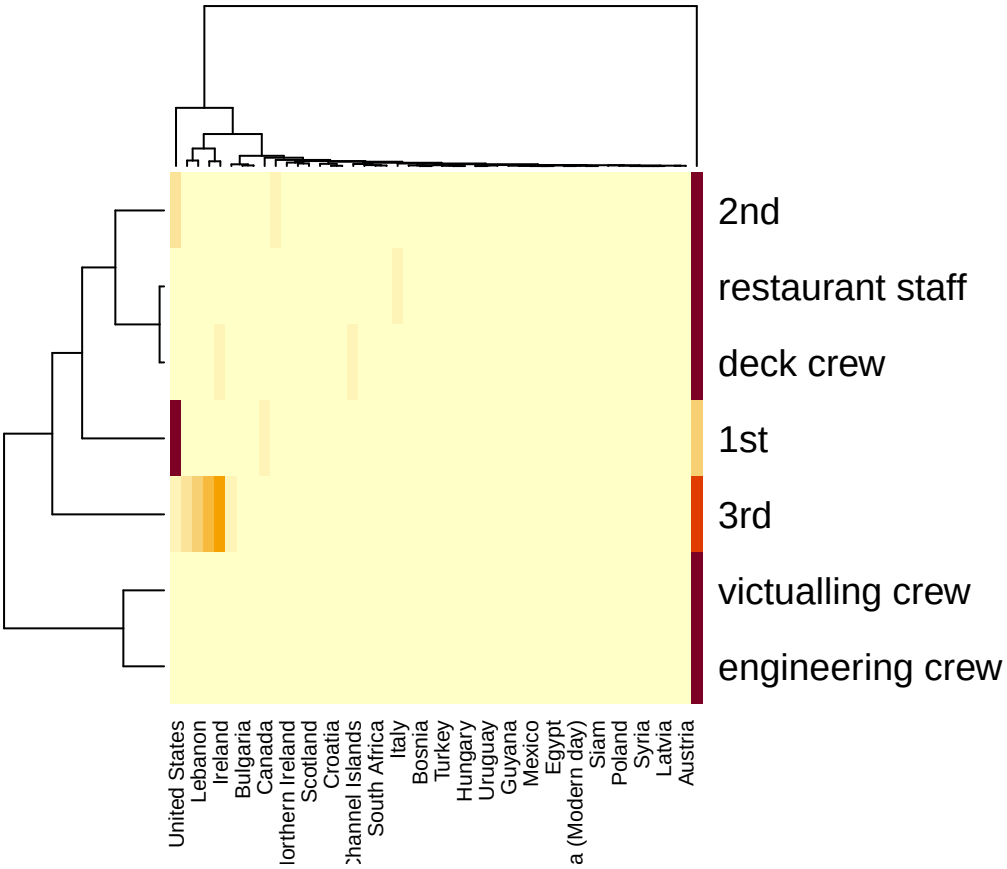


A continuación mostramos los mapas de calor generados que comparan la correlación entre la clase y el puerto de embarque así como el país de origen.

```
heatmap(data_cor_class_embarked)
```

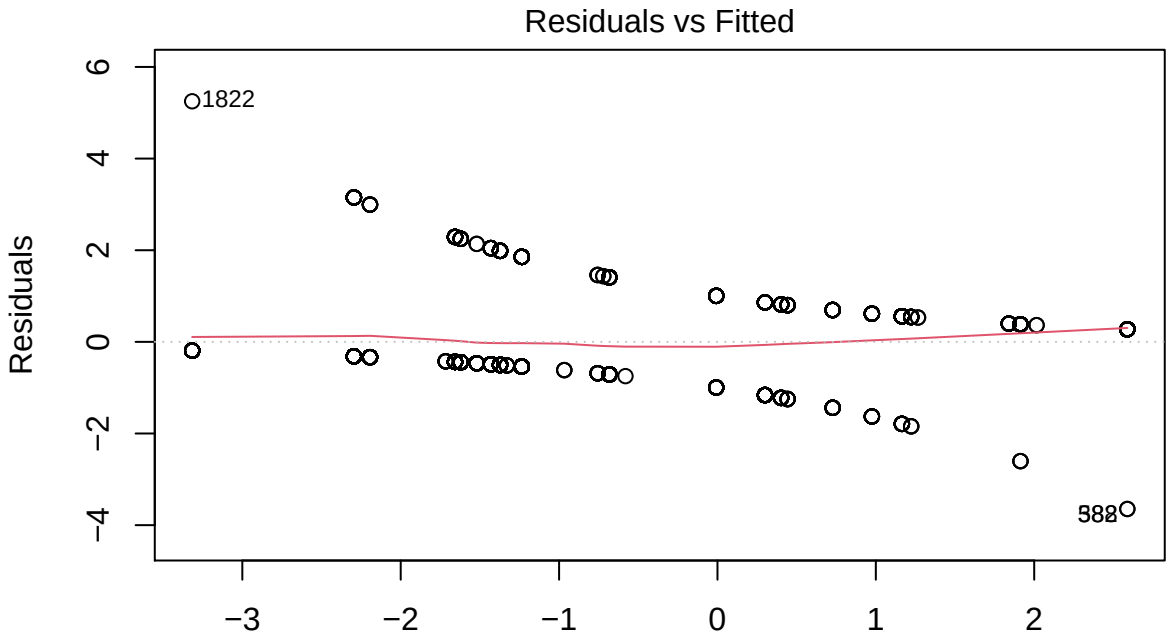


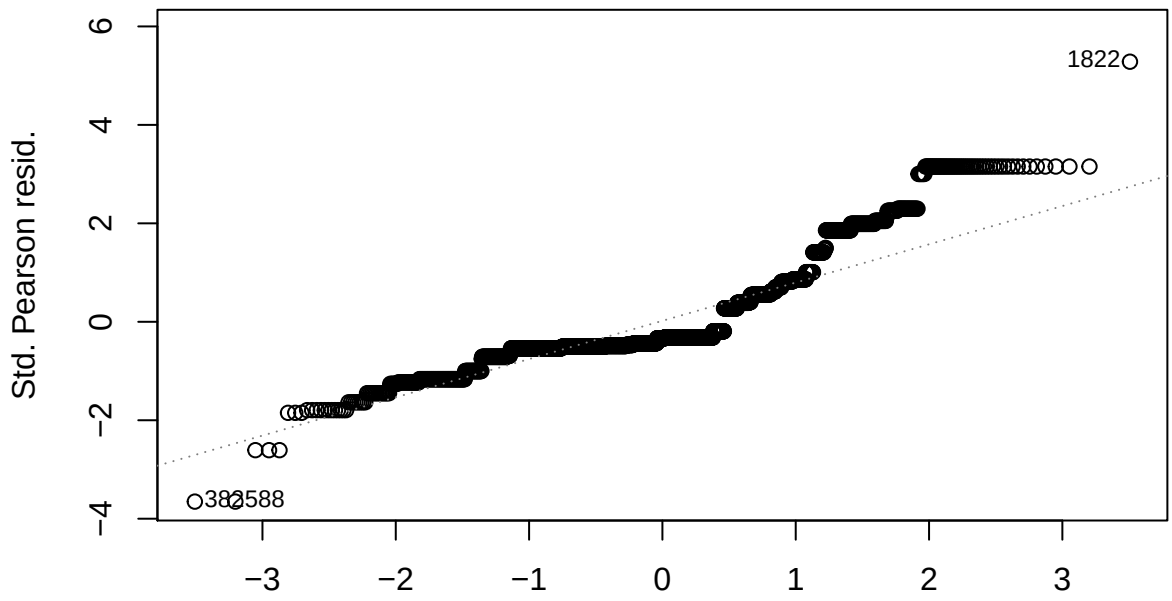

```
heatmap(data_cor_class_country)
```

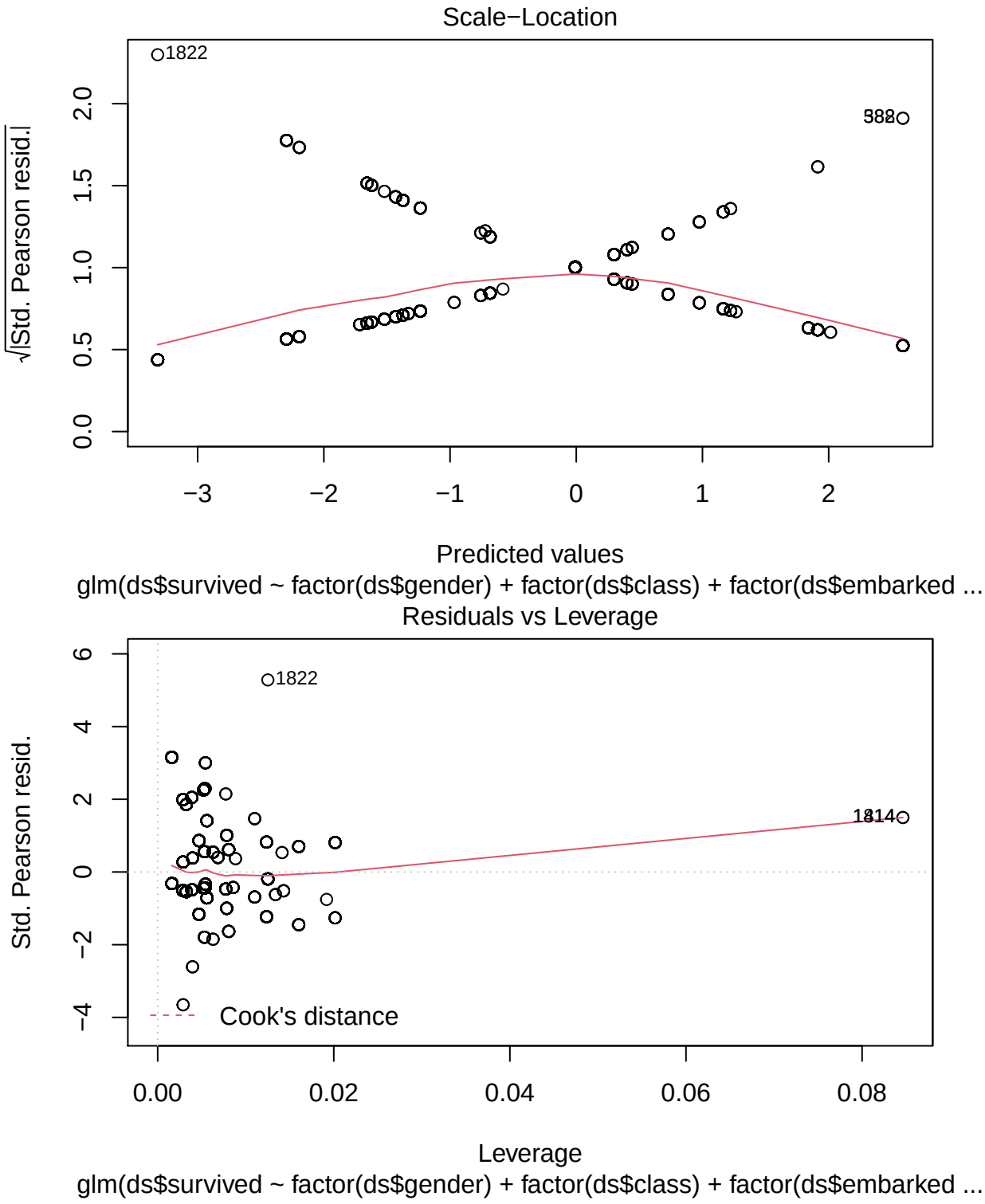


También se visualiza el resultado del modelo que mejor se comporta para los datos.

```
plot(model_gender_class_embarked)
```


Residuals vs Fitted
Residuals
1822
Predicted values
glm(ds$survived ~ factor(ds$gender) + factor(ds$class) + factor(ds$embarked ...


Normal Q−Q
Std. Pearson resid.
1822
Theoretical Quantiles
glm(ds$survived ~ factor(ds$gender) + factor(ds$class) + factor(ds$embarked ...



6. Resolución del problema.

A partir de los resultados obtenidos y de las gráficas mostradas en el apartado anterior podemos afirmar lo siguiente:

- Las variables que influyen de una forma estadísticamente significativa en las contrucción del modelo y por lo tanto en la supervivencia en el desastre histórico son `gender`, `class` y `embarked`.
- Las variables `age`, `sibsp`, `parch` y `family` no tienen gran impacto en la supervivencia.
- Existe una correlación entre el puerto de embarque **C** y ser de primera clase **1st**.
- Se ve claramente una correlación entre el puerto de embarque en **B** y formar parte de la tripulación **victualling** y **deck**. Aunque está siendo sesgada por la elevada proporción de pasajeros que inician el viaje en **S**.
- Se encuentra que la mayoría de individuos de primera clase son estadounidenses.

7. Exportar dataset resultante

```
write.csv(ds, "./data/titanic_processed.csv", row.names = T)
```

8. Tabla contribuciones

Contribuciones	**Firma**
Investigación Previa	Eleazar Morales Díaz, Susana Vila Melero
Redacción de las respuestas	Eleazar Morales Díaz, Susana Vila Melero
Desarrollo del código	Eleazar Morales Díaz, Susana Vila Melero