Práctica 2: Limpieza y validación de los datos

Autores: Mejía Quintero Dayana, Peterson Christopher

Junio 2022

Contents

COMPETENCIAS DE LA PRACTICA

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science: - Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo. - Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

OBJETIVOS DE LA PRÁCTICA

Los objetivos concretos de esta práctica son: - Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares. - Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpiezay validación) para llevar a cabo un proyecto analítico. - Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos. - Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico. - Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación. - Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo. - Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

DESCRIPCIÓN DEL DATASET

El dataset escogido representa a las personas que embarcaron en el titanic, el cual naufragó en el Océano del Atlántico Norte el 15 de abril de 1912 al ser impactado por un iceberg. Dado de que vamos a usar dos datasets: Train y test, vamos a unificarlos en un solo dataset para proceder a la limpieza de datos y posterior análisis. El fichero contiene: 1309 observaciones de 12 variables que anteriormente se mencionaron en la descripción del dataset.

El dataset final contiene las siguientes variables:

- PassengerID: ID del pasajero.
- Survival: Sobreviviente. Está compuesto con 0=No, 1= Yes.
- Name: Nombre de los pasajeros
- Pclass: Clase del tiquete. Está compuesto con 1 = 1era, 2= 2da, 3 = 3ra. En donde funciona como proxy del estatus socioeconómico. 1era = Clase alta, 2da= Clase Media, 3ra= Clase Baja.
- Sex: Sexo.
- Age: Edad en años. Donde Age es fraccional si es menor a 1. Si la edad es estimada, entonces tiene forma de xx.5.
- Sibsp: # de hermanos / pareja dentro del titanic. El dataset define a las relaciones familiares como Sibling= hermano, hermana, hermanastro, hermanastra. Pareja = esposa, esposo (amantes y prometidos fueron ignorados).
- Parch: # de padres/hijos dentro del titanic. El dataset define las relaciones familiares como: Parent = madre, padre. Child = Hija, hijo, hijastra, hijastro. Algunos niños viajaron solo con su niñera, por lo tanto, parch= 0 para ellos.
- Ticket: Número del tiquete.
- Fare: Tarifa del pasajero.
- Cabin: Número de la cabina.
- Embarked: Puerto de embarcación. Está compuesto por C= Cherbourg, Q = Queenstown, S = Southampton.

IMPORTANCIA Y OBJETIVOS DEL ANÁLISIS DEL DATASET.

Hemos escogido el dataset relacionado con las personas que embarcaron el titanic que se encuentra en la página kaggle: https://www.kaggle.com/competitions/titanic/data en la cual separa los datos en dos datasets: test y train. Donde train.csv contiene los detalles de un subset de los pasajeros a bordo del titanic los cuales son 891 en total y en donde se revelará si sobreviven o no. El test.csv contiene información similar, pero con ella debemos predecir cuál de estas condiciones sucede. El objetivo de esta práctica es predecir si los pasajeros a bordo sobreviven o no y también encontrar si ciertas variables como la pclass que identifica la clase del tiquete influenciaron en la sobrevivencia de los pasajeros y si otras variables entraron como a influenciar de forma más impredecible.

INTEGRACIÓN Y SELECCIÓN DE LOS DATOS A ANALIZAR.

Es importante escoger las variables que consideramos importantes que nos ayudarán en el proceso del análisis del dataset para posteriormente llegar a los objetivos planteados en esta práctica. Dichas variables deben contener la información más relevante que nos ayude llegar a dicho paso y resolver el problema planteado. Al observar el dataset y ver como se comporta las variables podemos reducir la dimensionalidad y también reducir el dataset, eliminando las variables que consideramos que no ayudan a la resolución. En nuestro caso, se eliminará las siguientes variables:

- PassengerID: ID del pasajero.
- Name: Nombre de los pasajeros
- Ticket: Número del tiquete.
- Cabin: Número de la cabina.

Instalamos los paquetes

Instalamos y cargamos las librarías requeridas.

```
if (!require(GGally)) install.packages(GGally); library(GGally)
```

```
## Loading required package: GGally
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
if (!require('rpart.plot')) install.packages('rpart.plot'); library('rpart.plot')
```

```
## Loading required package: rpart.plot
```

```
## Loading required package: rpart
```

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('rpart')) install.packages('rpart'); library('rpart')
if (!require('randomForest')) install.packages('randomForest'); library('randomForest')
```

```
## Loading required package: randomForest

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
if (!require("dplyr")) install.packages("dplyr"); library("dplyr")
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':
##
##     combine

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

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

```
tinytex::install_tinytex()
```

Dado de que vamos a usar dos datasets: Train y test, vamos a unificarlos en un solo dataset para proceder luego a la limpieza de datos y posterior análisis. El fichero unificado contiene: 1309 observaciones de 12 variables que anteriormente se mencionaron en la descripción del dataset. Por ahora mantendremos todas las variables para su observación total.

```
test <- read.csv('test.csv',stringsAsFactors = FALSE)
train <- read.csv('train.csv', stringsAsFactors = FALSE)

# Creamos un nuevo dataset con ambos archivos como se había mencionado anteriormente.
df <- bind_rows(train,test)
len_train=dim(train)[1]
```

Hacemos una rápida observación del dataset donde vemos el número de variables y el número de observaciones que ya se ha mencionado. También se puede ver las caracteristicas de las variables del dataset.

```r
str(df)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ 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" ...
```

Observamos las estadisticas principales de las variables:

```r
summary(df)
```

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

Eliminamos variables que no vamos a utilizar:

```r
#Eliminamos PassengerId, Name, Ticket, Cabin,
dfaux<-df[,c("Survived","Pclass","Sex","Age","Fare","SibSp","Parch", "Embarked")]
```

Vemos como quedo:

```
summary(dfaux);
```

```
##     Survived          Pclass          Sex                 Age
##  Min.   :0.0000   Min.   :1.000   Length:1309        Min.   : 0.17
##  1st Qu.:0.0000   1st Qu.:2.000   Class :character   1st Qu.:21.00
##  Median :0.0000   Median :3.000   Mode  :character   Median :28.00
##  Mean   :0.3838   Mean   :2.295                      Mean   :29.88
##  3rd Qu.:1.0000   3rd Qu.:3.000                      3rd Qu.:39.00
##  Max.   :1.0000   Max.   :3.000                      Max.   :80.00
##  NA's   :418                                         NA's   :263
##       Fare             SibSp            Parch          Embarked
##  Min.   :  0.000   Min.   :0.0000   Min.   :0.000   Length:1309
##  1st Qu.:  7.896   1st Qu.:0.0000   1st Qu.:0.000   Class :character
##  Median : 14.454   Median :0.0000   Median :0.000   Mode  :character
##  Mean   : 33.295   Mean   :0.4989   Mean   :0.385
##  3rd Qu.: 31.275   3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :512.329   Max.   :8.0000   Max.   :9.000
##  NA's   :1
```

PROCESO DE LIMPIEZA DE LOS DATOS.

Ahora procederemos a la limpieza de los datos analizando los valores vacios, nulos y los valores extremos u outliers.

Eliminación de valores nulos y vacios.

Para comenzar en la limpieza de datos, observamos las variables que contienen valores vacios la cual la razón suele ser porque no se llego a registrar la información.

```
# Valores vacios
colSums(is.na(dfaux))
```

```
## Survived   Pclass      Sex      Age     Fare    SibSp    Parch Embarked
##      418        0        0      263        1        0        0        0
```

```
colSums(dfaux=="")
```

```
## Survived   Pclass      Sex      Age     Fare    SibSp    Parch Embarked
##       NA        0        0       NA       NA        0        0        2
```

Como podemos ver, las variables “Age” y la variable “Embarked” contiene valores vacios. Existen diferentes metodos para poder solucionar este problema. En nuestro caso, utilizaremos el método de reemplazo con la media para dichos valores. En el caso de “Embarked” utilizaremos que vamos a reemplzar los valores vacios con la primera opción que es “S”

```
# Tenemos muchos valores vacios en "Age" por lo que lo reemplazaremos con la media.
dfaux$Age[is.na(dfaux$Age)] <- mean(dfaux$Age, na.rm = TRUE)

# Cambiamos los valores vacios de Embarked por la primera opción que es "S"
dfaux$Embarked[dfaux$Embarked==""]="S"
```

Identificación y gestión de valores extremos u outliers.

Los valores extremos u outliers son aquellos valores que se encuentran alejados del resto de observaciones y pueden llegar a ser valores tanto muy pequeños o muy grandes. Para su analisis es necesario tambien comprender las razones del porque se pueden generar este tipo de valores para no eliminarlos y sesgar el analisis afectando el modelo. Utlizaremos la herramienta de boxplot.stats para identificar dichos valores.

Primero vamos a convertir las variables de Survived, Pclass, Sex, Embarked a factores dado de que estos valores toman valores finitos:

```
# Convertimos los datos de Survived, Pclass, Sex, Embarked a factores

dfaux$Survived <- as.factor(dfaux$Survived)
dfaux$Pclass <- as.factor(dfaux$Pclass)
dfaux$Sex <- as.factor(dfaux$Sex)
dfaux$Embarked <- as.factor(dfaux$Embarked)
```

Veamos las variables con los valores extremos usando boxplot.stats.

Comenzamos para Edad “Age”.

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

```
##   [1]  2.00 58.00 55.00  2.00 66.00 65.00  0.83 59.00 71.00 70.50  2.00 55.50
##  [13]  1.00 61.00  1.00 56.00  1.00 58.00  2.00 59.00 62.00 58.00 63.00 65.00
##  [25]  2.00  0.92 61.00  2.00 60.00  1.00  1.00 64.00 65.00 56.00  0.75  2.00
##  [37] 63.00 58.00 55.00 71.00  2.00 64.00 62.00 62.00 60.00 61.00 57.00 80.00
##  [49]  2.00  0.75 56.00 58.00 70.00 60.00 60.00 70.00  0.67 57.00  1.00  0.42
##  [61]  2.00  1.00 62.00  0.83 74.00 56.00 62.00 63.00 55.00 60.00 60.00 55.00
##  [73] 67.00  2.00 76.00 63.00  1.00 61.00 60.50 64.00 61.00  0.33 60.00 57.00
##  [85] 64.00 55.00  0.92  1.00  0.75  2.00  1.00 64.00  0.83 55.00 55.00 57.00
##  [97] 58.00  0.17 59.00 55.00 57.00
```

La variable “Pclass”

```
boxplot.stats(dfaux$Pclass)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors
```

```
## factor(0)
## Levels: 1 2 3
```

La variable “Sex”

```
boxplot.stats(dfaux$Sex)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors
```

```
## factor(0)
## Levels: female male
```

La variable "Embarked"

```
boxplot.stats(dfaux$Embarked)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors
```

```
## factor(0)
## Levels: C Q S
```

La variable "Fare"

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

```
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000
##   [9] 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000  66.6000  69.5500
##  [17]  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000  83.4750  90.0000
##  [25]  79.2000  86.5000 512.3292  79.6500 153.4625 135.6333  77.9583  78.8500
##  [33]  91.0792 151.5500 247.5208 151.5500 110.8833 108.9000  83.1583 262.3750
##  [41] 164.8667 134.5000  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000
##  [49] 263.0000  75.2500  69.3000 135.6333  82.1708 211.5000 227.5250  73.5000
##  [57] 120.0000 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042  91.0792
##  [65]  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000 221.7792 106.4250
##  [73]  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833  79.6500  79.2000
##  [81]  78.2667 153.4625  77.9583  69.3000  76.7292  73.5000 113.2750 133.6500
##  [89]  73.5000 512.3292  76.7292 211.3375 110.8833 227.5250 151.5500 227.5250
##  [97] 211.3375 512.3292  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583
## [105] 211.3375  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500
## [113]  89.1042 164.8667  69.5500  83.1583  82.2667 262.3750  76.2917 263.0000
## [121] 262.3750 262.3750 263.0000 211.5000 211.5000 221.7792  78.8500 221.7792
## [129]  75.2417 151.5500 262.3750  83.1583 221.7792  83.1583  83.1583 247.5208
## [137]  69.5500 134.5000 227.5250  73.5000 164.8667 211.5000  71.2833  75.2500
## [145] 106.4250 134.5000 136.7792  75.2417 136.7792  82.2667  81.8583 151.5500
## [153]  93.5000 135.6333 146.5208 211.3375  79.2000  69.5500 512.3292  73.5000
## [161]  69.5500  69.5500 134.5000  81.8583 262.3750  93.5000  79.2000 164.8667
## [169] 211.5000  90.0000 108.9000
```

La variable "Survived"

```
boxplot.stats(dfaux$Survived)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors
```

```
## factor(0)
## Levels: 0 1
```

La variable “SibSp”

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

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

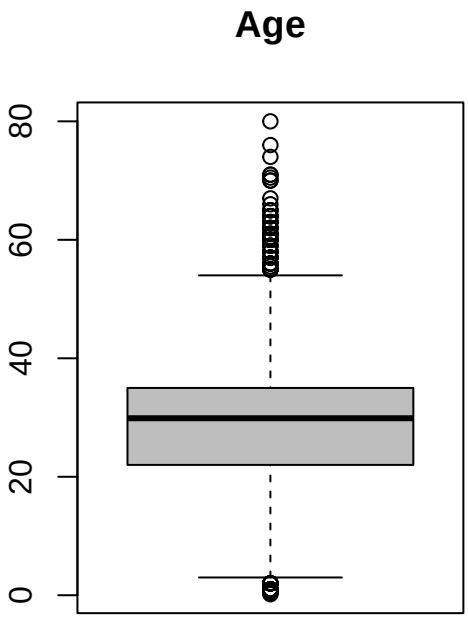
La variable “Parch”

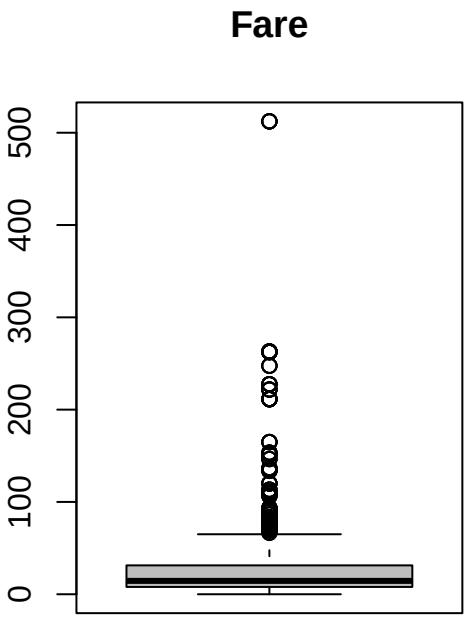
```
boxplot.stats(dfaux$Parch)$out
```

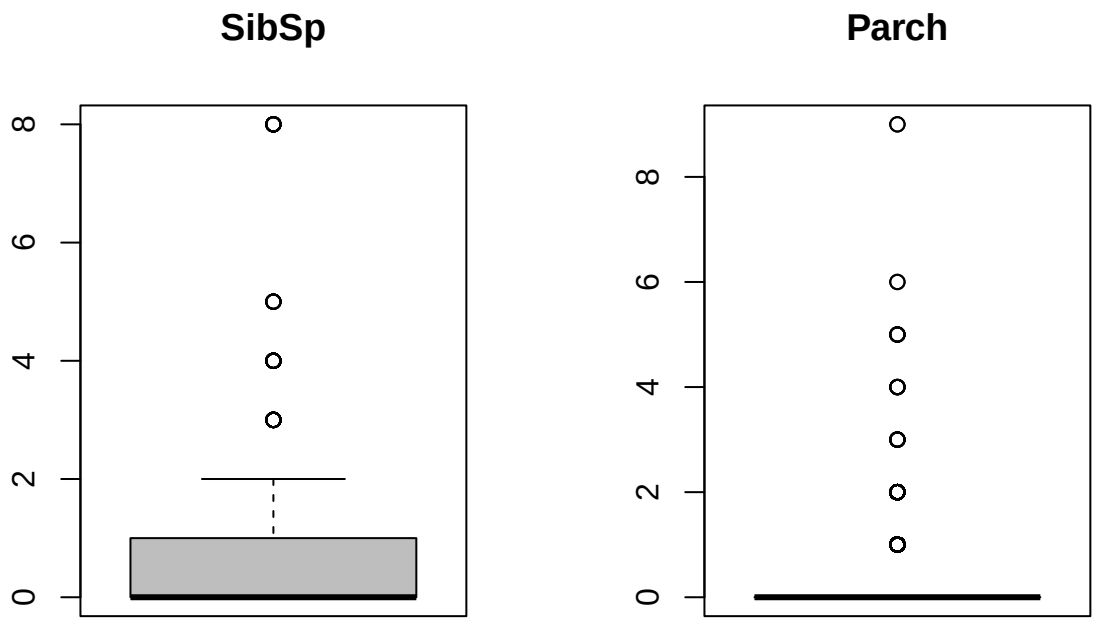
```
##   [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1 2 1
##  [38] 1 2 1 4 1 1 1 1 2 2 1 2 1 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 1 1 1 1 1 2 1 2
##  [75] 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2 2 3 4 1 2 1
## [112] 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1 2 1 1 2 5 2 1 1
## [149] 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1 1 2 1 2 3 1 2 1 2 2
## [186] 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1 1 5 2 1 1 1 1 3 1 2 2 1
## [223] 2 1 2 1 2 4 1 1 2 1 1 1 4 6 2 3 1 1 2 2 2 1 1 2 5 2 3 2 1 1 1 2 1 2 2 2 1
## [260] 2 1 1 2 1 2 1 2 1 2 2 1 1 1 1 1 2 1 1 2 1 1 1 2 1 2 9 1 1 1 2 2 2 1 9 1 1
## [297] 2 2 1 1 2 1 1 1 1 1 1
```

Utlizando un diagrama de caja nos dara lo siguiente en forma gráfica:

```
outliers <- function(dfaux) {
    par(mfrow=c(1,2))
    for(i in 1:ncol(dfaux)) {
        if (is.numeric(dfaux[,i])){
            boxplot(dfaux[,i], main = colnames(dfaux)[i], width = 100, col="gray")
        }
    }

    max(dfaux$Age, na.rm = TRUE)
    min(dfaux$Age, na.rm = TRUE)
    fivenum(dfaux$Age)

    max(dfaux$Fare, na.rm = TRUE)
    min(dfaux$Fare, na.rm = TRUE)
    fivenum(dfaux$Fare)
}
outliers(dfaux)
```


Age
0
20
40
60
80


Fare
0
100
200
300
400
500




```
## [1]   0.0000   7.8958  14.4542  31.2750 512.3292
```

De lo anterior podemos observar que la variable edad tiene valores extremos pero hay que tener en cuenta que las edades comprendidas entre 60 años y 80 años son normales, tambien que una persona tenga 0.92 años ya que representa a un bebe. . Tambien en el caso de Fare, existen tambien valores extremos pero de acuerdo a la cabina comprada puede ser normal dicho precio gastado por los pasajeros. Esto en resumen nos lleva a que no quitaremos los valores extremos porque podemos asegurarnos que son validos dado las condiciones de las variables.

Ahora, eliminamos las variables que nos vamos a utilizar:

ANÁLISIS DE LOS DATOS.

Analizaremos el dataset ya limpio para observar como se comportan las variables.

Selección de grupos de datos a analizar.

Agrupamos los datos en grupos:

Agrupamos los datos que se quieren comparar

```
# Por Cabina
dfaux.first_class <- dfaux[dfaux$Pclass == 1,]
dfaux.second_class <- dfaux[dfaux$Pclass == 2,]
```

```r
dfaux.third_class <- dfaux[dfaux$Pclass == 3,]

print(paste("First_class: ", nrow(dfaux.first_class)))
```

```
## [1] "First_class:  323"
```

```r
print(paste("Second_class: ", nrow(dfaux.second_class)))
```

```
## [1] "Second_class:  277"
```

```r
print(paste("Third_class: ", nrow(dfaux.third_class)))
```

```
## [1] "Third_class:  709"
```

```r
# Por genero
dfaux.male <- dfaux[dfaux$Sex == "male",]
dfaux.female <- dfaux[dfaux$Sex == "female",]

print(paste("Male: ", nrow(dfaux.male)))
```

```
## [1] "Male:  843"
```

```r
print(paste("Female: ", nrow(dfaux.female)))
```

```
## [1] "Female:  466"
```

```r
# Por Embarque
dfaux.C <- dfaux[dfaux$Embarked == "C",]
dfaux.Q <- dfaux[dfaux$Embarked == "Q",]
dfaux.S <- dfaux[dfaux$Embarked == "S",]

print(paste("Cherbourg: ", nrow(dfaux.C)))
```

```
## [1] "Cherbourg:  270"
```

```r
print(paste("Queenstown: ", nrow(dfaux.Q)))
```

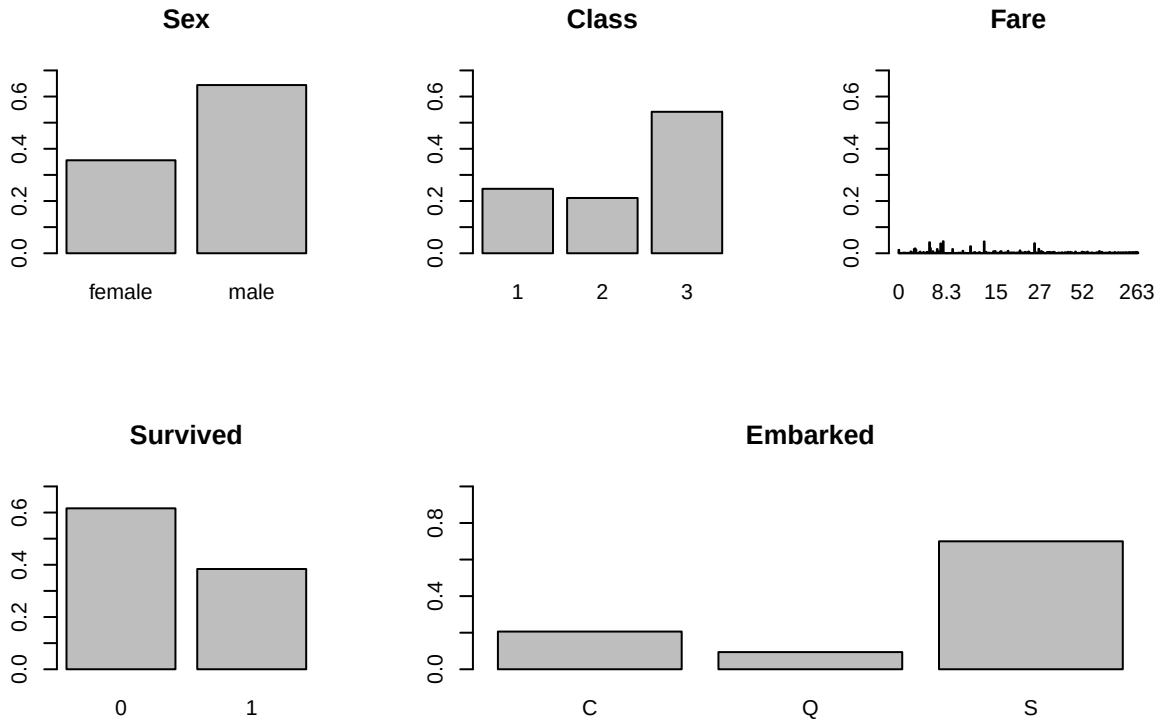
```
## [1] "Queenstown:  123"
```

```r
print(paste("Southampton: ", nrow(dfaux.S)))
```

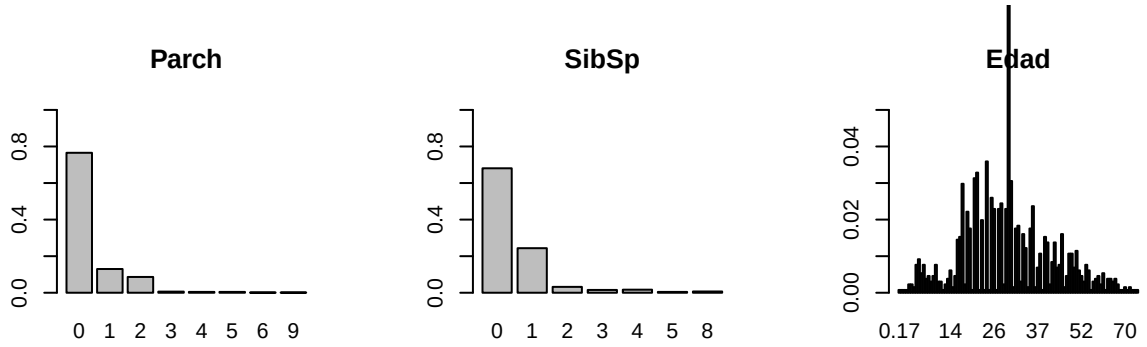
```
## [1] "Southampton:  916"
```

Primero vamos a graficar las frecuencias usando un barplot con todas las variables y observar como se comportan:

```
# Gráfica de las Frecuencias de cada una de las variables del dataset
dataaux<-layout(matrix(c(1,2,3,4,5,5), 2, 3, byrow=TRUE),respect=TRUE);
barplot(prop.table(table(dfaux$Sex)),ylim=c(0,0.7), main="Sex");
barplot(prop.table(table(dfaux$Pclass)),ylim=c(0,0.7), main="Class");
barplot(prop.table(table(dfaux$Fare)),ylim=c(0,0.7), main="Fare");
barplot(prop.table(table(dfaux$Survived)),ylim=c(0,0.7), main="Survived");
barplot(prop.table(table(dfaux$Embarked)),ylim=c(0,1), main="Embarked");
```




```
barplot(prop.table(table(dfaux$Parch)),ylim=c(0,1), main="Parch");
barplot(prop.table(table(dfaux$SibSp)),ylim=c(0,1), main="SibSp");
barplot(prop.table(table(dfaux$Age)),ylim=c(0,0.05), main="Edad");
```

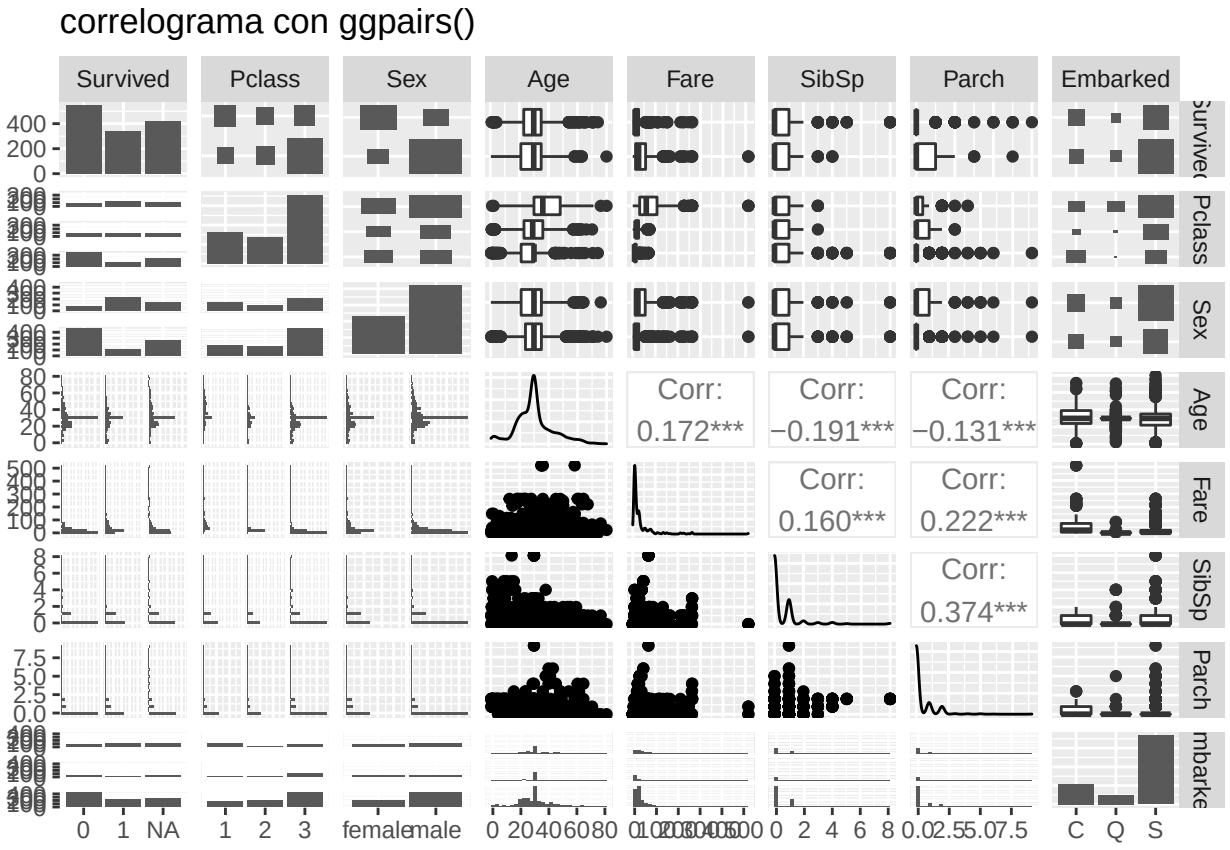


Ahora vamos a observamos las correlaciones de ellas usando varias grñaficas como el scatterplot, las distribuciones y el coeficiente de correlación.

```
# Vemos las correlaciones (usando scatterplots), distribuciones e imprimimos el coeficiente de correlac
ggpairs(dfaux, title="correlograma con ggpairs()")
```

```
## Warning: Removed 418 rows containing non-finite values (stat_g_gally_count).
## Removed 418 rows containing non-finite values (stat_g_gally_count).

## Warning: Removed 418 rows containing missing values (stat_boxplot).
## Removed 418 rows containing missing values (stat_boxplot).
## Removed 418 rows containing missing values (stat_boxplot).
## Removed 418 rows containing missing values (stat_boxplot).

## Warning: Removed 418 rows containing non-finite values (stat_g_gally_count).

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
## Removed 1 rows containing non-finite values (stat_boxplot).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (stat_bin).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (stat_bin).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (stat_bin).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing missing values (geom_point).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing missing values (geom_point).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (stat_bin).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Comprobación de la normalidad y homogeneidad de la varianza.

Ya que hemos anterioremente agrupado las variables, podemos realizar la comprobación de la homogeneidad con la función de Fligner- Killeen la cual es un test no parametrico para la homogeneidad de un grupo de varianzas basada en rangos.

```
varianza <- fligner.test(dfaux);
varianza;
```

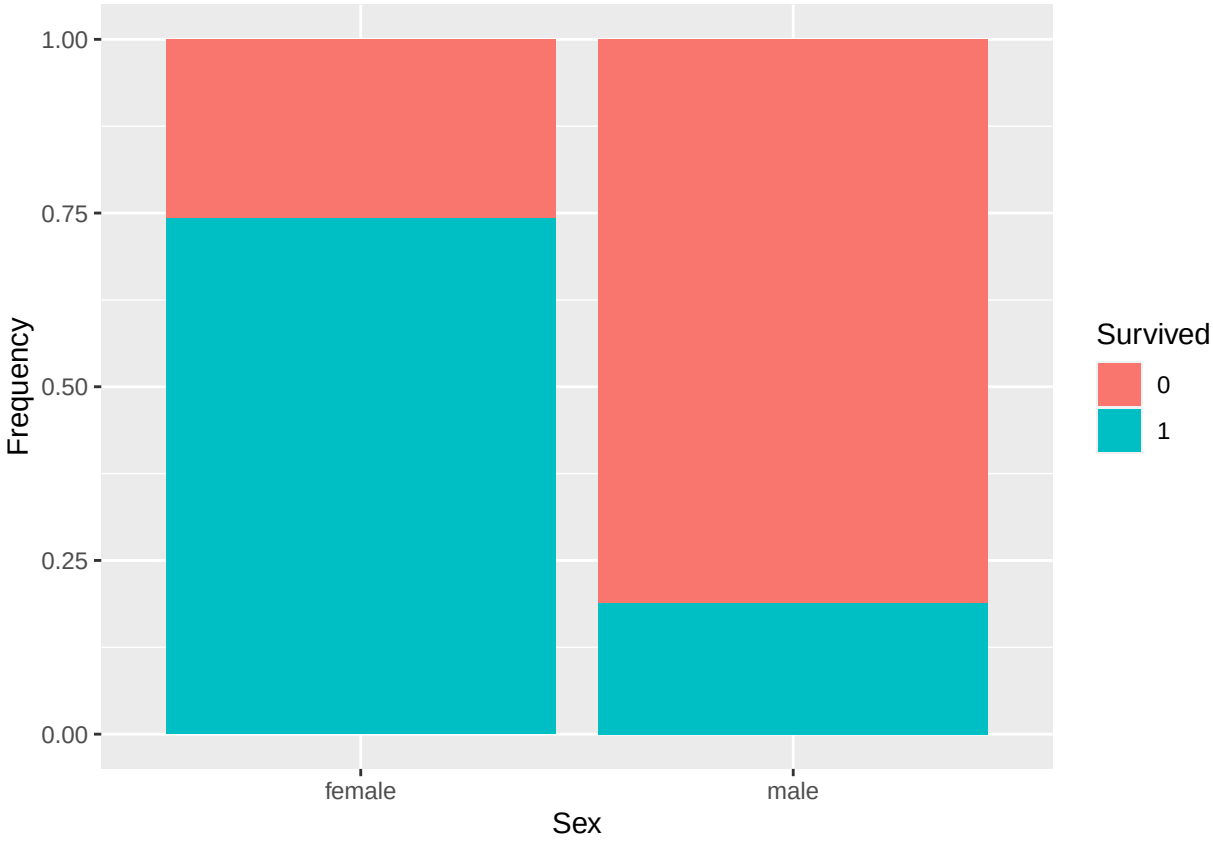
```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  dfaux
## Fligner-Killeen:med chi-squared = 5407.1, df = 7, p-value < 2.2e-16
```

Esto nos trae como conclusión que p-value $<$ 2.2e-16 lo que las varianzas de las variables son diferentes.

Ahora, en el caso de la normalidad, vamos a hacer el analisis con dos variables, es decir, survived, y otra variable.

Relación entre Sex y survival Podemos observar que alrededor del 75% de las mujeres fueron sobrevivientes del naufragio mientras que hay un procentaje menor de sobrevivientes hombres con menos del 23%.

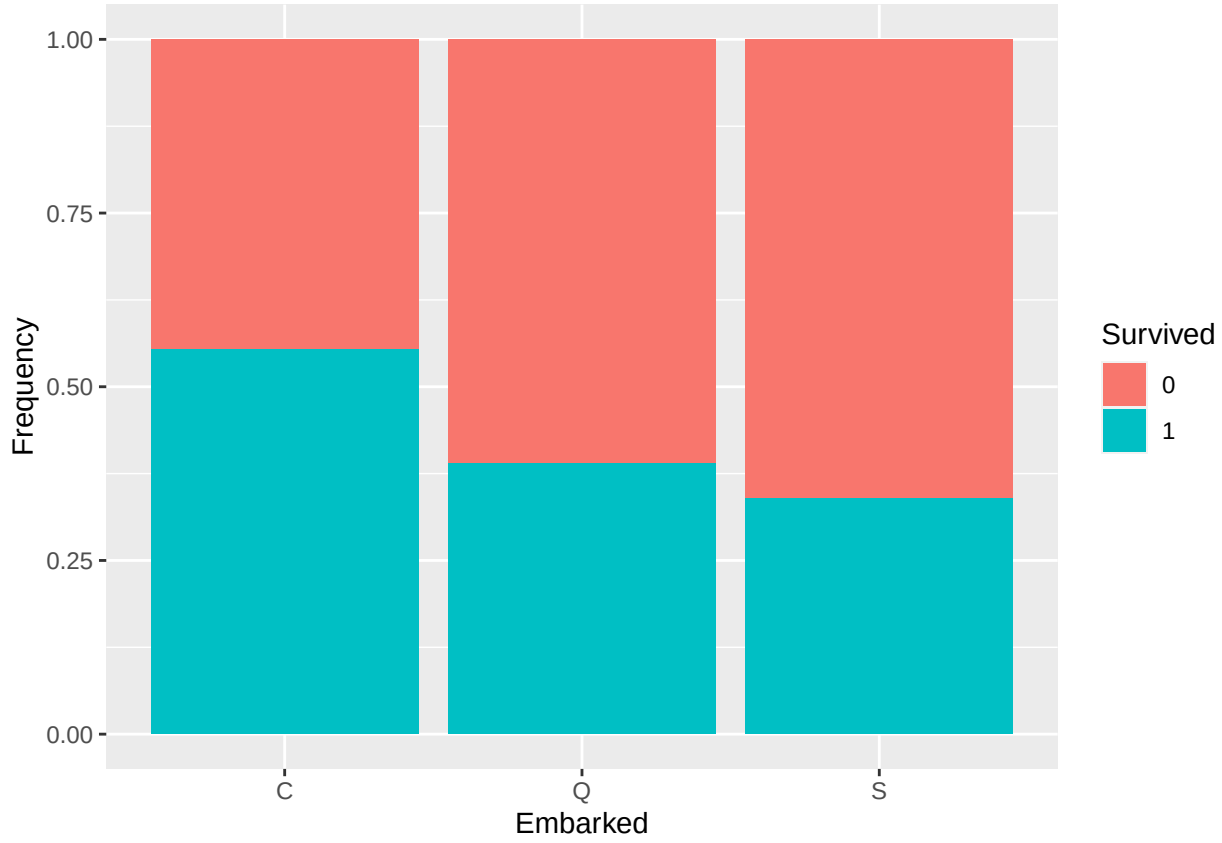
```r
# Comprobación de la normalidad.
#Vamos a ver la relación entre sex y survival.
ggplot(data=dfaux[1:len_train,],aes(x=Sex,fill=Survived))+geom_bar(position="fill")+ylab("Frequency")
```



Relación entre Embarked y survival

Los pasajeros que más sobrevivieron fueron los que los embarcaron desde Cherbourg con alrededor del 56%, alrededor del 38% que arrivaron en Queenstown sobrevivieron, y alrededor del 35% que embarcaron en S = Southampton sobrevivieron.

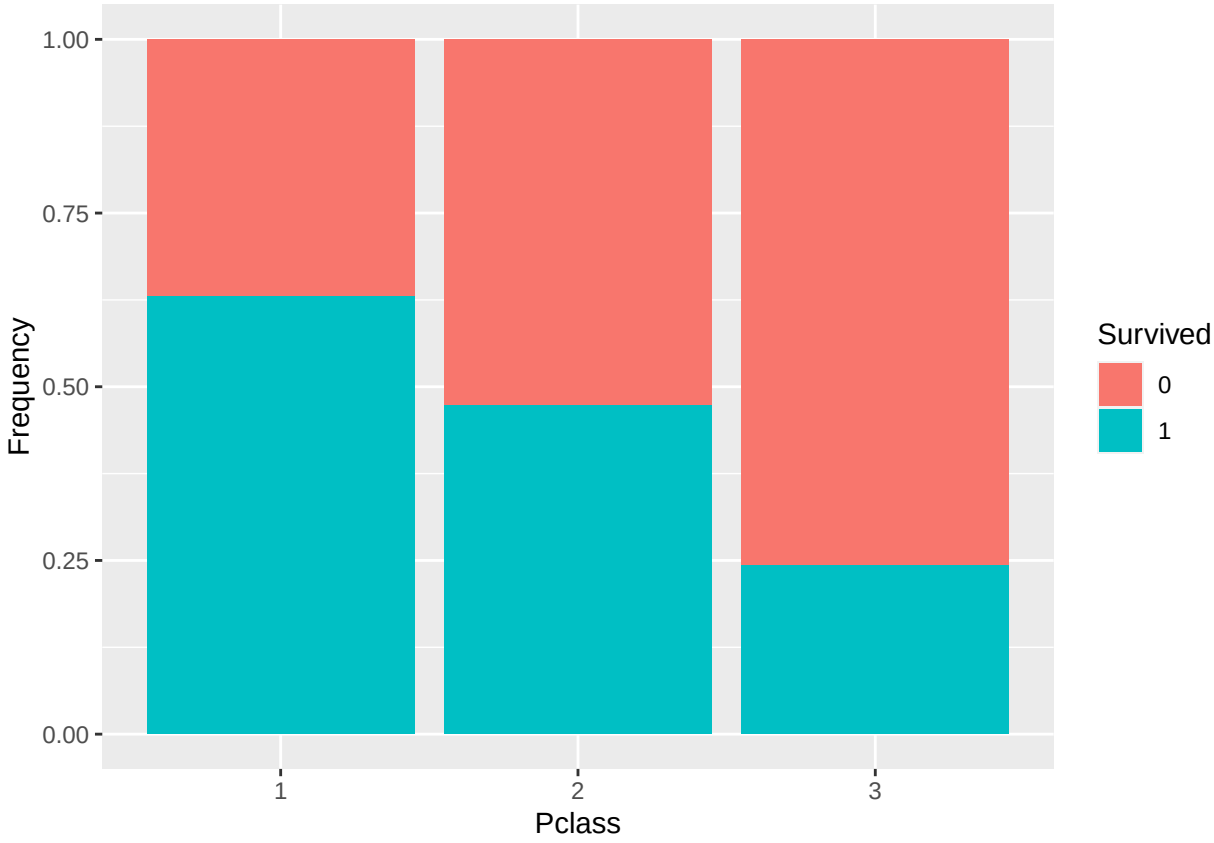
```r
# Survival como función de embarked:
ggplot(data = dfaux[1:len_train,],aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+ylab("Freque
```



Relación entre PClass y survival

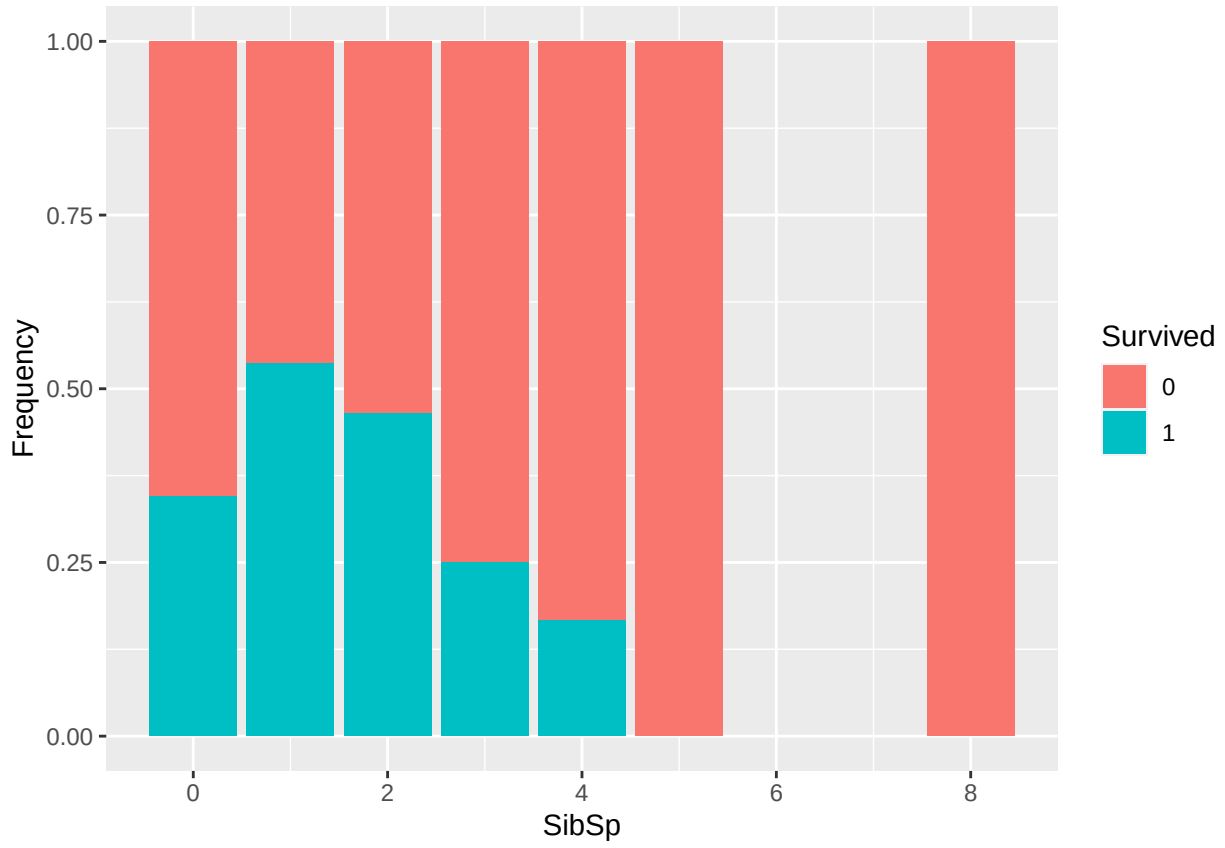
Podemos observar que alrededor de la clase 1 el 63% fueron sobrevievientes, De la clase 2, alrededor del 48% sobrevivieron y de la clase 3 del 25% sobrevivieron.

```
# Survival como función de Pclass:
ggplot(data = dfaux[1:len_train,],aes(x=Pclass,fill=Survived))+geom_bar(position="fill")+ylab("Frequenc
```



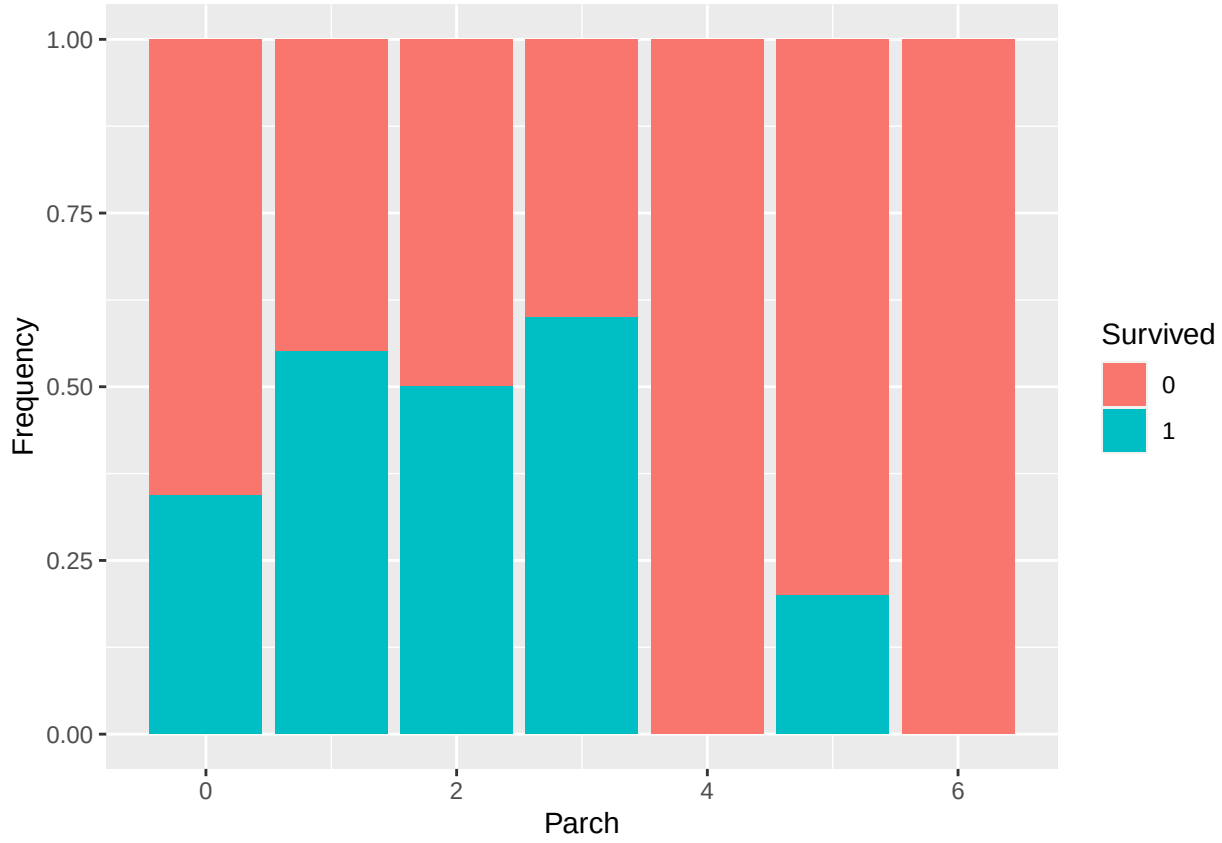
Relación entre SibSp y survival

```
# Survivial as a function of SibSp
ggplot(data = dfaux[1:len_train,],aes(x=SibSp,fill=Survived))+geom_bar(position="fill")+ylab("Frequency
```



Relación entre Parch y survival

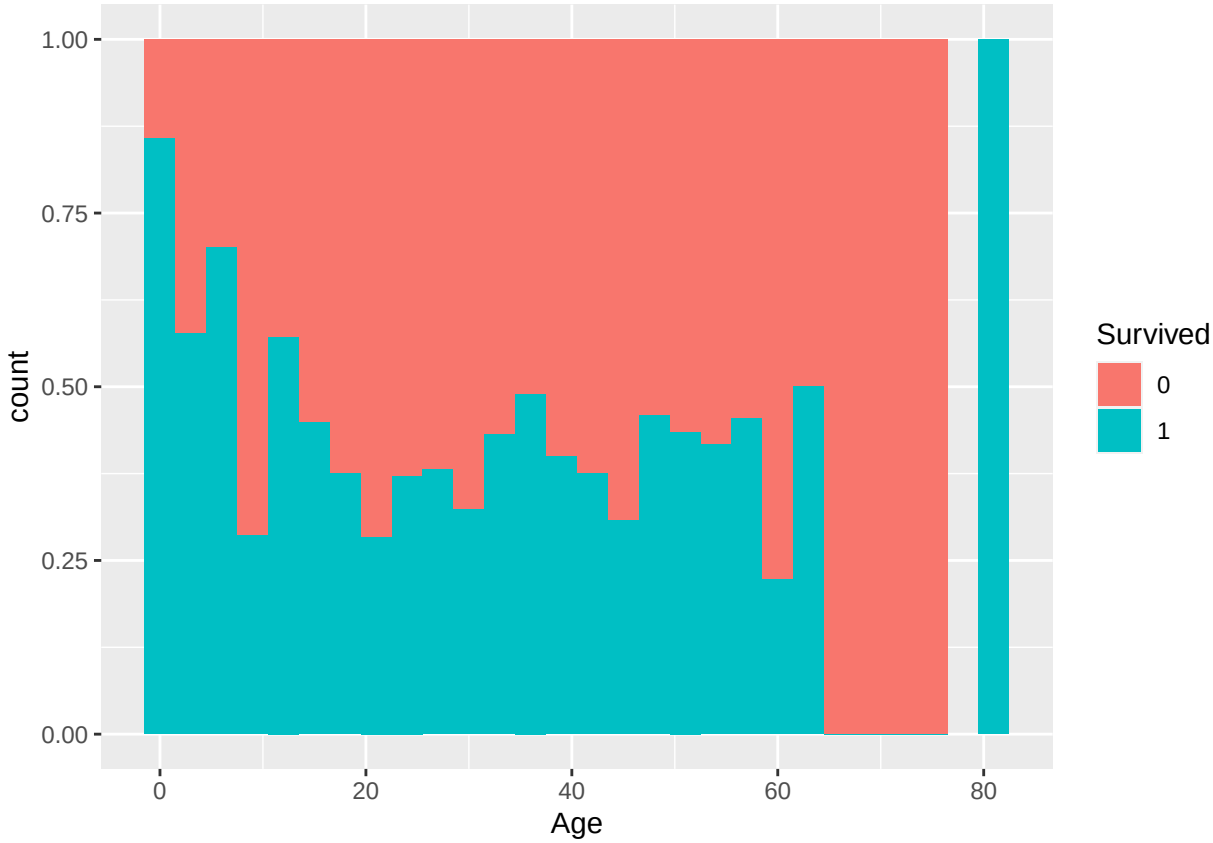
```
# Survivial como función de  Parch
ggplot(data = dfaux[1:len_train,],aes(x=Parch,fill=Survived))+geom_bar(position="fill")+ylab("Frequency
```



Relación entre Age y survival

```
# Survival as a function of age:

ggplot(data = dfaux[1:len_train,],aes(x=Age,fill=Survived))+geom_histogram(binwidth =3, position="fill"
```

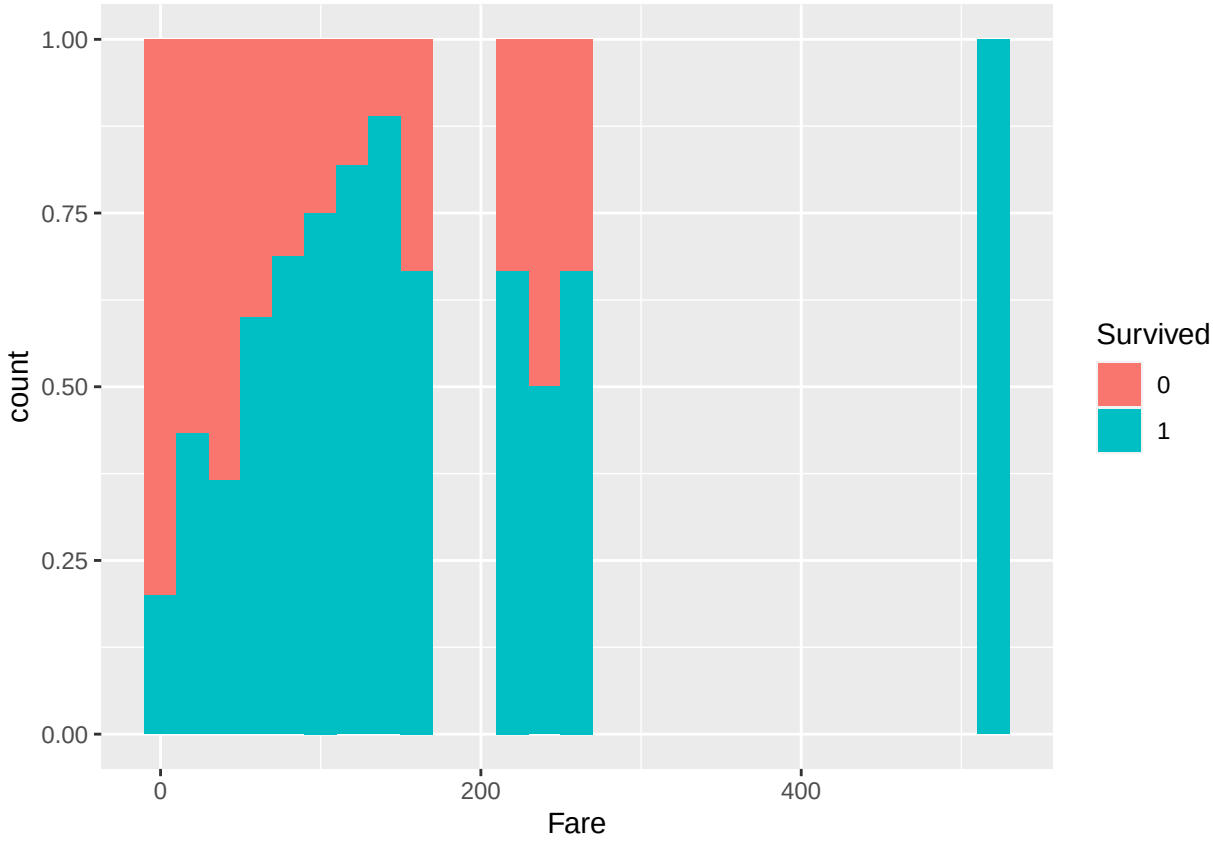
```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



Relación entre Fare y survival

```
# Correlación entre Fare y Survival
ggplot(data = dfaux[1:len_train,],aes(x=Fare,fill=Survived))+geom_histogram(binwidth =20, position="fil
```

```
## Warning: Removed 28 rows containing missing values (geom_bar).
```



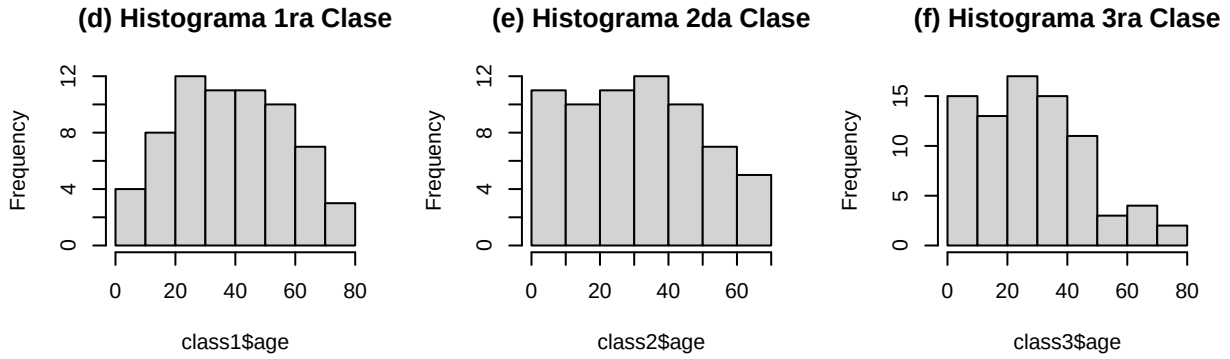
Si vemos la influencia de la edad, el sexo y la clase en los sobrevivientes tendriamos:

Anteriormente habiamos hecho subset de las variables como agrupación de datos pero ahora lo haremos con relación al Pclass. Primero agrupamos por pclass:

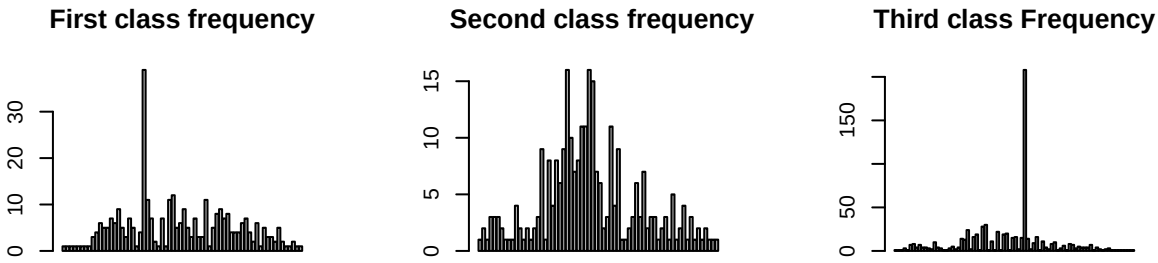
```
# Agregamos las variables que queremos estudiar que influyen la sobrevivencia como sex y age, con Pclass
sex_tot=aggregate(dfaux$Pclass, by=list(sex=dfaux$Sex, pclass=dfaux$Pclass), FUN=function(x){NROW(x)});
Pclass_tot=aggregate(dfaux$Pclass, by=list(pclass=dfaux$Pclass), FUN=function(x){NROW(x)});
age_tot=aggregate(dfaux$Pclass, by=list(age=dfaux$Age, pclass=dfaux$Pclass), FUN=function(x){NROW(x)});

# Hacemos un subset basado en los valores de sex

men<-subset(sex_tot, sex=='male');
women<-subset(sex_tot, sex=='female');

# Ahora vemos el porcentaje de hombres y mujeres.
men$percentage <- round(prop.table(men$x),4)*100;
women$percentage <- round(prop.table(women$x),4)*100;

#Sacamos el subset de edad basado en Pclass.
class1<-subset(age_tot, pclass=='1');
class2<-subset(age_tot, pclass=='2');
class3<-subset(age_tot, pclass=='3');
```

```
# Vamos a graficar lo anterior en relación a la pclass. Utilizaremos el histrograma, el barplot y el qqp
data1<-layout(matrix(c(1,2,3,4,5,6), 2, 3, byrow=TRUE),respect=TRUE);
```

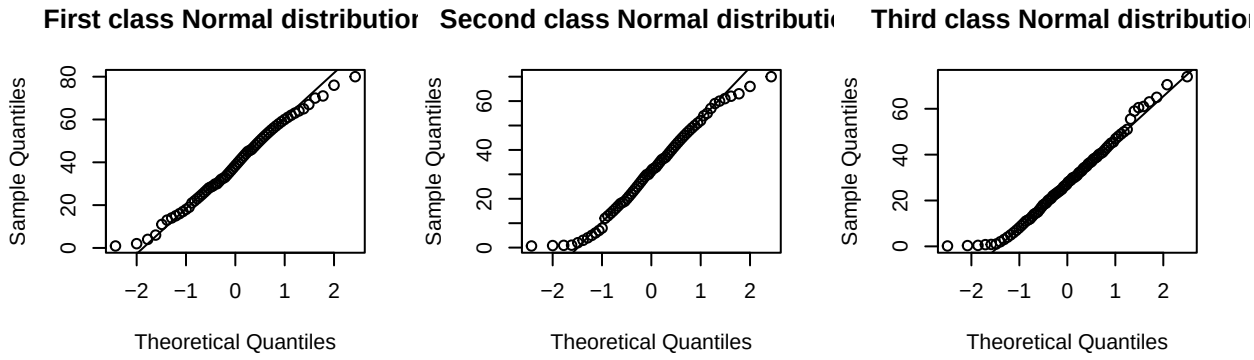
```r
hist(class1$age, main="(d) Histograma 1ra Clase");
hist(class2$age, main="(e) Histograma 2da Clase");
hist(class3$age, main="(f) Histograma 3ra Clase");
```




```r
# Usando el barplot
data1<-layout(matrix(c(1,2,3,4,5,6), 2, 3, byrow=TRUE),respect=TRUE);

barplot(class1$x, main="First class frequency");
barplot(class2$x, main=" Second class frequency");
barplot(class3$x, main="Third class Frequency");
```




```
# Usando el qqnorm con qqline
data1<-layout(matrix(c(1,2,3,4,5,6), 2, 3, byrow=TRUE),respect=TRUE);

qqnorm(class1$age, main="First class Normal distribution");
qqline(class1$age);
qqnorm(class2$age, main="Second class Normal distribution");
qqline(class2$age);
qqnorm(class3$age, main= "Third class Normal distribution");
qqline(class3$age);
```



Con los gráficos anteriores podemos encontrar que poseen una distribución normal.

APLICACIÓN DE PRUEBAS ESTADISTICAS.

Vamos a aplicar ahora pruebas estadisticas para observar cuales son las variables que influyen más en la sobrevivencia de los pasajeros y cual seria la sobrevivencia de los pasajeros aplicando modelos. En nuestro caso, utilizaremos de nuevo nuestros dataset del train y del test. Aquí aplicaremos el metodo del arbol de decisión pero primero realizaremos la regresión logistica.

```
# Seleccionamos de nuevo los datos de train y test y escogemos las variables que vamos a utilizar
train<-dfaux[1:len_train,c("Survived","Pclass","Sex","Age","Fare","SibSp","Parch", "Embarked")]

len_test<-dim(test)[1]
test<-tail(dfaux,len_test)
test<-test[,c("Survived","Pclass","Sex","Age","Fare","SibSp","Parch", "Embarked")]


# Hacemos un regresion logistica

model <- glm(Survived ~.,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
```

```
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6271  -0.6093  -0.4218   0.6173   2.4497
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.108317   0.476722   8.618  < 2e-16 ***
## Pclass2     -0.932800   0.297867  -3.132  0.00174 **
## Pclass3     -2.156069   0.297799  -7.240 4.49e-13 ***
## Sexmale     -2.718678   0.201099 -13.519  < 2e-16 ***
## Age         -0.039136   0.007872  -4.972 6.64e-07 ***
## Fare         0.002292   0.002469   0.928  0.35325
## SibSp       -0.323596   0.109731  -2.949  0.00319 **
## Parch       -0.097449   0.119052  -0.819  0.41305
## EmbarkedQ   -0.025521   0.382000  -0.067  0.94673
## EmbarkedS   -0.440410   0.239742  -1.837  0.06621 .
## ---
## 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:  784.29  on 881  degrees of freedom
## AIC: 804.29
##
## Number of Fisher Scoring iterations: 5
```

Vamos a realizar la predicción de los sobrevievientes con nuestro modelo con el dataset del train.

```
# Ahora, vemos la prediccion de los sobrevivientes

pred.train <- predict(model,train)
pred.train <- ifelse(pred.train > 0.5,1,0)

# Media de la prediccion verdadera
mean(pred.train==train$Survived)
```

```
## [1] 0.8136925
```

```
t1<-table(pred.train,train$Survived)
# precisión y recall del modelo
precision<- t1[1,1]/(sum(t1[1,]))
recall<- t1[1,1]/(sum(t1[,1]))
precision
```

```
## [1] 0.799687
```

```
recall
```

```
## [1] 0.9307832
```

```
# F1 score
F1<- 2*precision*recall/(precision+recall)
F1
```

```
## [1] 0.8602694
```

Vamos a ver el accuracy de nuestro modelo.

```
table(train$Survived, pred.train >= 0.5)
```

```
##
##     FALSE TRUE
##   0   511   38
##   1   128  214
```

```
accuracy = (244 + 458) / nrow(train)
sensitivity = 244 / (244 + 98)
specificity = 458 / (458 + 91)

cat("accuracy: ", accuracy)
```

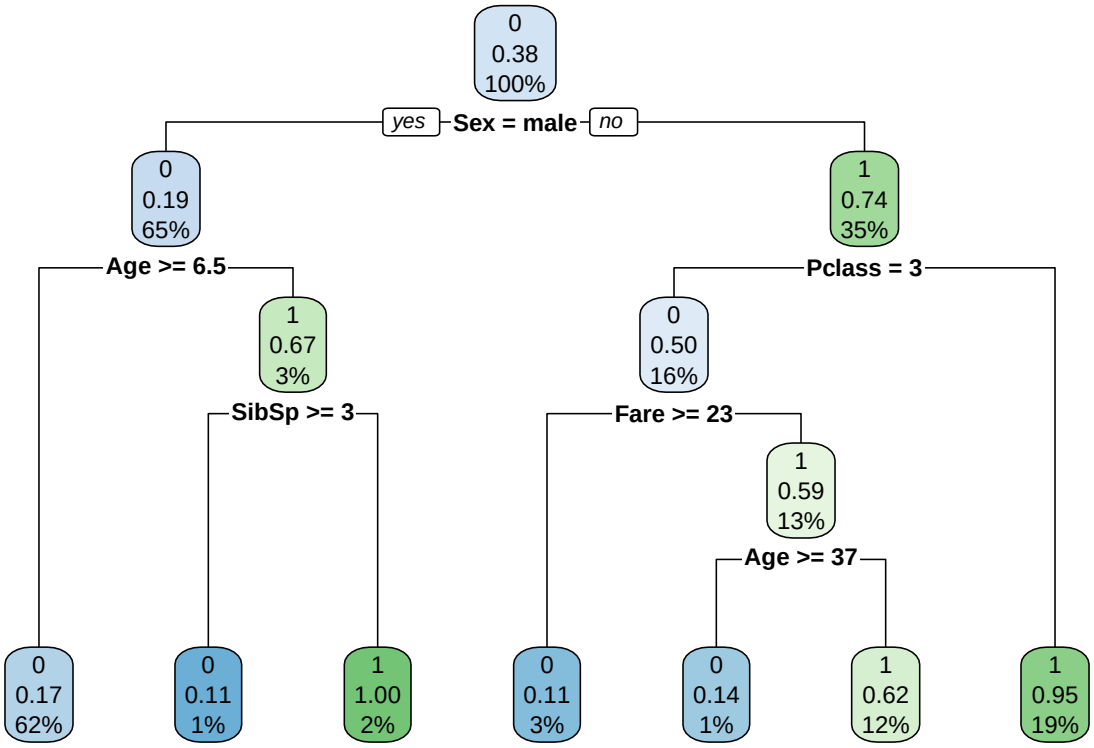
```
## accuracy:  0.7878788
```

REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS A PARTIR DE TABLAS Y GRÁFICAS

Existen diferentes gráficas para usar como modelos de prediccion. Uno de ellos es el arbol de decisión el cual se busca predecir la probabilidad de que se llegue el objetivo en base a ciertas condiciones. Usamos en nuestro caso para representar los resultados, un arbol de decisión el cual al comienzo transforma la variable survived para que a partir de sexo puede ir ramificandose la resolución al problema.

Gráficamos con rpart.plot el arbol de desición donde se puede varios niveles.

```
# Representamos el arbol de decision
model_dt<- rpart(Survived ~.,data=train, method="class")
rpart.plot(model_dt)
```



Como quedaría el arbol:

```
# Vemos como queda el arbol
model_dt
```

```
## n= 891
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 891 342 0 (0.61616162 0.38383838)
##    2) Sex=male 577 109 0 (0.81109185 0.18890815)
##      4) Age>=6.5 553  93 0 (0.83182640 0.16817360) *
##      5) Age< 6.5 24   8 1 (0.33333333 0.66666667)
##       10) SibSp>=2.5 9   1 0 (0.88888889 0.11111111) *
##       11) SibSp< 2.5 15   0 1 (0.00000000 1.00000000) *
##    3) Sex=female 314  81 1 (0.25796178 0.74203822)
##      6) Pclass=3 144  72 0 (0.50000000 0.50000000)
##       12) Fare>=23.35 27   3 0 (0.88888889 0.11111111) *
##       13) Fare< 23.35 117  48 1 (0.41025641 0.58974359)
##         26) Age>=36.5 7   1 0 (0.85714286 0.14285714) *
##         27) Age< 36.5 110  42 1 (0.38181818 0.61818182) *
##      7) Pclass=1,2 170   9 1 (0.05294118 0.94705882) *
```

Podemos tambien exportar los resultados de la predicción en un archivo csv.

```r
pred.test <- predict(model,test)
pred.test <- ifelse(pred.test > 0.5,1,0)

pred.test
```

```
##  892  893  894  895  896  897  898  899  900  901  902  903  904  905  906  907
##    0    0    0    0    0    0    1    0    1    0    0    0    1    0    1    1
##  908  909  910  911  912  913  914  915  916  917  918  919  920  921  922  923
##    0    0    0    0    0    0    1    1    1    0    1    0    0    0    0    0
##  924  925  926  927  928  929  930  931  932  933  934  935  936  937  938  939
##    0    0    0    0    0    1    0    0    0    0    0    1    1    0    0    0
##  940  941  942  943  944  945  946  947  948  949  950  951  952  953  954  955
##    1    0    0    0    1    1    0    0    0    0    0    1    0    0    0    1
##  956  957  958  959  960  961  962  963  964  965  966  967  968  969  970  971
##    1    1    1    0    0    1    1    0    0    0    1    1    0    1    0    1
##  972  973  974  975  976  977  978  979  980  981  982  983  984  985  986  987
##    0    0    0    0    0    0    1    1    1    0    0    0    1    0    0    0
##  988  989  990  991  992  993  994  995  996  997  998  999 1000 1001 1002 1003
##    1    0    1    0    1    0    0    0    1    0    0    0    0    0    0    1
## 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019
##    1    1    1    0    0    1    0    1    1    0    1    0    0    1    0    0
## 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035
##    0    0    0    0    0    0    0    0    0    0    1    0    0    1    0    0
## 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051
##    0    0    0    0    0    0    1    0   NA    0    0    0    1    1    0    0
## 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067
##    1    0    1    0    0    0    0    0    1    1    0    0    0    0    0    1
## 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083
##    1    0    1    1    0    0    1    0    1    0    1    0    0    0    0    0
## 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099
##    0    0    0    0    1    1    0    0    1    0    0    1    0    0    1    0
## 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115
##    1    0    0    0    0    0    0    0    1    0    1    0    1    0    1    0
## 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131
##    1    1    0    1    0    0    0    1    0    0    0    0    0    0    1    1
## 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147
##    1    1    0    0    0    0    1    0    1    0    1    0    0    0    0    0
## 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163
##    0    0    1    0    0    0    1    1    0    0    0    0    0    0    0    0
## 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179
##    1    0    0    1    0    0    0    0    1    0    1    1    1    0    0    0
## 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195
##    0    0    0    1    0    0    0    0    1    0    0    0    0    0    0    0
## 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211
##    1    1    0    0    0    0    0    0    0    0    1    1    0    0    0    0
## 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227
##    0    0    0    0    1    0    1    0    0    0    1    0    0    1    0    0
## 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243
##    0    0    0    0    0    0    0    1    0    1    0    0    0    1    1    0
## 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259
##    0    0    1    0    1    0    0    0    0    1    1    0    1    0    0    1
## 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275
##    1    0    0    1    0    0    1    1    0    0    0    0    0    0    0    0
```

```
## 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291
##    0    1    0    0    0    0    0    1    0    0    0    1    0    1    0    0
## 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307
##    1    0    1    0    0    0    0    0    1    1    1    1    0    0    1    0
## 1308 1309
##    0    0
```

```
test$Survived<- pred.test

write.csv(pred.test,file="prediction.csv",row.names = F)
```

Tambien se exporta el archivo ya limpio.

```
write.csv(dfaux,file="archivolimpio.csv",row.names = F)
```

RESOLUCIÓN DE PROBLEMAS Y CONCLUSIONES

Con el uso de arboles de decisión para la resolución de nuestro problema en cuanto a la sobrevivencia de los pasajeros y el impacto de ciertas variables en ella, hemos encontrado que nos trae los siguientes resultados comprendidos en varios niveles:

- si es hombre (male) y tiene una edad Age>=6.5 entonces muere
- Si es hombre (male) y tiene una edad Age< 6.5 y SibSp>=2.5 entonces muere
- Si es hombre (male) y tiene una edad Age< 6.5 y SibSp<2.5 entonces sobrevive
- si es mujer (female) y pclass< 3 entonces sobrevive
- si es mujer (female), Pclass=3, Fare>=23.35, entonces muere
- si es mujer (female), Pclass=3, Fare< 23.35, Age>=36.5 entonces muere
- si es mujer (female), Pclass=3, Fare< 23.35, Age< 36.5 entonces sobrevive

Se puede observar que vcon estos resultados llegamos a la conclusión de que no hay influencia de las clases en los hombres en donde hay prioridad a los niños menores de 6.5 años y que tienen una cantidad menor de hermanos. En el caso de los hombres, la supervivencia es muy pequeña a diferencia de las mujeres. En el caso de las mujeres, vemos que las clase 1 y 2 que son consideradas alta y media, sobreviven, los que nos demuestra que juega un papel importante el aspecto socioeconomico en la supervivencia. En el caso de las mujeres con la pclass 3, que es la baja, entra los factores en donde cuando son menores de 36.5 años sobreviven. De aquí podría comprobarse la teoría de mujeres y niños/as primero son los que más probabilidades tuvieron de sobrevivir en el titanic. Se observa tambien que la variable **pclass** tiene una influencia mucho mayor que las otras variables como anteriormente se había mencionado.

EXPORTACIÓN DEL CÓDIGO

En el github se puede observar el archivo subido con el código realizado en el programa r. Tambien se encuentra en el github los archivos csv de los dataset, el dataset limpio y el de las predicciones.

- Mejia Quintero Dayana: https://github.com/danamejia1810/Practica-2-Titanic-dataset.git
- Peterson Christopher: https://github.com/christopherapeterson/Practica-2-Titanic-dataset.git

BIBLIOGRAFÍA

- Calvo M, Subirats L, Pérez D (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
- Dalgaard, Peter (2008). Introductory statistics with R. Springer Science & Business Media.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Osborne, Jaso W. (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
- Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
- Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.