# Actividad: PRA2: Limpieza y análisis de datos

*Reison A. Torres Urina*

*Diciembre 2019*

## Índice



# 1. Detalles de la actividad

## 1.1. Descripción

En esta practica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

## 1.2. Objetivos

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, limpieza y 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.

### 1.3. Competencias

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.

## 2. Solución

### 2.1. Descripción del dataset

#### 2.1.1. Carga de los datos

Cargamos el conjunto de datos que se encuentran en los archivos **train.csv**, **test.csv** y **gender_submission.csv** en formato CSV, y representan los datos de los pasajeros que abordaron el Titanic.

Estos datos estarán representados en R por un dataframe para facilitar la manipulación de los mismos en nuestro análisis.

```
# cargamos paquetes R que vamos a utilizar durante nuestro anlisis

if(!require(ggplot2)){
    #install.packages('ggplot2', repos='http://cran.us.r-project.org')
    library(ggplot2)
}

if(!require(ggpubr)){
    #install.packages('ggpubr', repos='http://cran.us.r-project.org')
    library(ggpubr)
}

library(dplyr)
#library(Hmisc)
#library(corrplot)

# Carga del dataset contenido en el archivo train.csv
titanic.train <- read.csv("../datos/train.csv",stringsAsFactors = FALSE, header=T, sep=",")
# Carga del dataset contenido en el archivo test.csv
titanic.test <- read.csv("../datos/test.csv",stringsAsFactors = FALSE, header=T, sep=",")
# Carga del dataset contenido en el archivo gender_submission.csv
titanic.test.survived <- read.csv("../datos/gender_submission.csv",stringsAsFactors = FALSE, header=T, s
```

#### 2.1.2. Descripción

Los datos seleccionados, fueron obtenidos del sitio de data science, **www.Kaggle.com**, en el encontramos una variedad de dataset Open Data. El conjunto de datos seleccionados para desarrollar esta actividad es

**Titanic: Machine Learning from Disaster**, en este dataset encontramos, los datos de los pasajeros, que abordaron el Titanic en su viaje inaugural.

Los datos de este dataset se encuentran divididos en dos archivos train.csv con 891 observaciones y test.cvs con 418 observaciones para un total de 1309. El conjunto de datos esta descripto por un conjunto de 12 variables. Las característica presenten en este dataset, nos permitirá cumplir los objetivos propuestos en esta actividad.

Variables contenidas en el dataset **train.csv**:

```
summary(titanic.train)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##    Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

Variables contenidas en el dataset **test.cvs**:

```
summary(titanic.test)
```

```
##   PassengerId         Pclass          Name              Sex
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##  NA's   :86
```

```
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:418         Length:418
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 35.627
##  3rd Qu.: 31.500
##  Max.   :512.329
##  NA's   :1
```

Variables contenidas en el dataset **gender_submission.csv**:

```
summary(titanic.test.survived)
```

```
##   PassengerId       Survived
##  Min.   : 892.0   Min.   :0.0000
##  1st Qu.: 996.2   1st Qu.:0.0000
##  Median :1100.5   Median :0.0000
##  Mean   :1100.5   Mean   :0.3636
##  3rd Qu.:1204.8   3rd Qu.:1.0000
##  Max.   :1309.0   Max.   :1.0000
```

Este dataset **gender_submission.csv** contiene la variable de survived, que luego utilizaremos para agregar al dataset **titanic.test**.

A continuación describimos el conjunto de variables que conforman este dataset:

- **PassengerId:** Número consecutivo que identifica al pasajero.
- **Name:** Nombre del pasajero.
- **Sex:** Define el xeso del pasajero.
- **pclass:** Nivel socioeconómico del pasajero (1st = Upper, 2nd = Middle, 3rd = Lower).
- **age:** Edad del pasajero en años.
- **sibsp:** Número de hermanos o cónyuges.
- **parch:** Número de hijos, padres.
- **ticket:** Número del boleto de abordaje.
- **fare:** Precio del boleto.
- **cabin:** Número de la cabina.
- **embarked:** Puerto de embarque (C = Cherbourg, Q = Queenstown, S = Southampton).
- **survived:** Pasajero superviviente (0 = No, 1 = Yes)

### 2.1.3. Importancia y objetivos de los análisis

A partir de este conjunto de datos se plantea la problemática de determinar qué variables influyen más en la supervivencia de un pasajero en el naufragio del Titanic. Además, se podrá proceder a crear modelos de regresión que permitan predecir si un pasajero sobrevive o no en función de sus características y contrastes de hipótesis que ayuden a identificar propiedades interesantes en las muestras que puedan ser inferidas con respecto a la población.

Este tipo de análisis pueden ser utilizados por las aseguradoras del sector turístico, para determinar el riesgo que puede tener un turista al viajar en los trasatlánticos. Y asi poder ofreser las cobertura del seguro.

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

### 2.2.1. Integración

Con el fin de tener una estructura de datos coherente y única que contenga mayor cantidad de información, combinaremos los datos procedentes de los dataset train.csv y test.cvs. Luego realizaremos una fusión horizontal para añadir el atributo **survived**, debido a que el dataset **test.cvs** no presenta este atributo. Este valor será extraído del dataset **gender_submission.csv**.

```
# Realizamos una fusión horizontal entre los dataset titanic.test y titanic.test.survived para agregar
titanic.test <- inner_join(titanic.test, titanic.test.survived, by ="PassengerId")

#Creamos el dataset titanic.data con la combinacion de los datos de los dataset  titanic.train y titani
titanic.data <- bind_rows(titanic.train,titanic.test)

# Eliminamos los dataset temporales
rm(titanic.test.survived)
rm(titanic.test)
rm(titanic.train)

# Verificamos la estructura del dataset con los datos combinados
summary(titanic.data)
```

```
##   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   :0.3774   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##
##      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
##  Length:1309        Min.   :  0.000   Length:1309
##  Class :character   1st Qu.:  7.896   Class :character
##  Mode  :character   Median : 14.454   Mode  :character
##                     Mean   : 33.295
##                     3rd Qu.: 31.275
##                     Max.   :512.329
##                     NA's   :1
##    Embarked
##  Length:1309
##  Class :character
##  Mode  :character
```

```
##
##
##
##
```

### 2.2.2. Selección de los datos

La gran mayoría de las variables contenidas en el conjunto de datos corresponde con características de los pasajeros que abordaron el Titanic, por lo que serán tenidas en cuenta para realizar nuestro análisis. Sin embargo, podremos prescindir de las variables (**PassengerId**,**Name** y **Ticket**) dado que estos atributos no aportan una carasterisitica al pasajero, y no influye en la resolución de nuestro problema.

```
# Eliminamos del dataset las variables "PassengerId" y "Name"
titanic.data <- titanic.data[,!(colnames(titanic.data) %in% c("PassengerId","Name","Ticket"))]

# Verificamos la estructura del dataset
summary(titanic.data)
```

```
##     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   :0.3774   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   :263
##      SibSp            Parch            Fare            Cabin
##  Min.   :0.0000   Min.   :0.000   Min.   :  0.000   Length:1309
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:  7.896   Class :character
##  Median :0.0000   Median :0.000   Median : 14.454   Mode  :character
##  Mean   :0.4989   Mean   :0.385   Mean   : 33.295
##  3rd Qu.:1.0000   3rd Qu.:0.000   3rd Qu.: 31.275
##  Max.   :8.0000   Max.   :9.000   Max.   :512.329
##                                   NA's   :1
##    Embarked
##  Length:1309
##  Class :character
##  Mode  :character
##
##
##
##
```

## 2.3. Limpieza de los datos

### 2.3.1. Discretización y conversion de tipos de datos

Al cargar los archivos con la función read.csv(), esta de manera automática asigna el tipo de variable en el dataset, en ciertas ocasiones los tipos asignados, no son los correctos. A continuación visualizamos los tipos de variables asignados al dataset, para luego decidir si se requiere una conversión de tipo.

```
# Tipos de variables
titanic.data.ctype <- sapply(titanic.data,class)
```

```
titanic.data.ctype <- data.frame(variables = names(titanic.data.ctype),tipo = as.vector(titanic.data.cty

titanic.data.ctype
```

```
##   variables      tipo
## 1  Survived   integer
## 2    Pclass   integer
## 3       Sex character
## 4       Age   numeric
## 5     SibSp   integer
## 6     Parch   integer
## 7      Fare   numeric
## 8     Cabin character
## 9  Embarked character
```

```
rm(titanic.data.ctype)
```

En este paso realizamos un analisis sobre las variables, que en R han sido cargadas como continuas pero en realidad son discretas (factor).Para esto realizamos un análisis de discretizacion sobre los atributos, para identificar que variables tienen sentido discretizar.

```
#summary(titanic.data[,titanic.data.ctype[titanic.data.ctype$tipo == "numeric",]$variables])
# Identificar el número de clases que se encuentra en cada variable del dataset
apply(titanic.data,2, function(x) length(unique(x)))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare    Cabin
##        2        3        2       99        7        8      282      187
## Embarked
##        4
```

Con el fin de facilitar la interpretar y comparar los resultados de diferentes grupos de datos, procedemos a discretizar a las variables con pocas clases:

```
cols<-c("Survived","Pclass","Sex","Embarked")
for (i in cols){
  titanic.data[,i] <- as.factor(titanic.data[,i]) # Conversion de variable a tipo factor
}

levels(titanic.data[,"Survived"]) <- c("No","Si")
levels(titanic.data[,"Pclass"]) <- c("Upper","Middle", "Lower")
levels(titanic.data[,"Embarked"]) <- c("?","Cherbourg", "Queenstown", "Southampton")

summary(titanic.data)
```

```
##  Survived    Pclass        Sex           Age            SibSp
##  No:815   Upper :323   female:466   Min.   : 0.17   Min.   :0.0000
##  Si:494   Middle:277   male  :843   1st Qu.:21.00   1st Qu.:0.0000
##           Lower :709                Median :28.00   Median :0.0000
##                                     Mean   :29.88   Mean   :0.4989
##                                     3rd Qu.:39.00   3rd Qu.:1.0000
##                                     Max.   :80.00   Max.   :8.0000
##                                     NA's   :263
##      Parch            Fare            Cabin                 Embarked
##  Min.   :0.000   Min.   :  0.000   Length:1309        ?          :  2
##  1st Qu.:0.000   1st Qu.:  7.896   Class :character   Cherbourg  :270
##  Median :0.000   Median : 14.454   Mode  :character   Queenstown :123
```

```
##  Mean   :0.385   Mean   : 33.295                    Southampton:914
##  3rd Qu.:0.000   3rd Qu.: 31.275
##  Max.   :9.000   Max.   :512.329
##                  NA's   :1
```

### 2.3.2. Tratamientos de ceros o elementos vacíos

Los datos vacíos o no definidos pueden presentarse en distintos formatos, típicamente “”, ? ,“ “ o NA (Not Available en inglés), pero en algunos contextos pueden incluso tomar valores numéricos como 0 o 999.

A continuación inspeccionaremos, que atributos de nuestro dataset, tienen una cantidad alta de valores no disponibles o valores faltantes en los diferentes formatos (“”,?,  o NA):

```
# Funcion: Explorar atributos con valores faltante
# Parmetros:
# 1. dataset: conjunto de datos con los atributos a explorar
hasValoresFaltantes <- function(dataset){
  # Verificar si existen variables cuantitativas con valores NA
  variablesWithNA <- colSums(is.na(dataset))

  # Verificar si existen variables con cadenas vacias
  variablesWithEmpaty <- colSums(dataset=="")
  variablesWithEmpaty[is.na(variablesWithEmpaty)] <- 0

  # Verificar si existen variables con valores desconocidos ("?").
  variablesWithQuestionMark <- colSums(dataset=="?")
  variablesWithQuestionMark[is.na(variablesWithQuestionMark)] <- 0

  # Verificar si existen variables con valores desconocidos (" ").
  variablesWithSpace <- colSums(dataset==" ")
  variablesWithSpace[is.na(variablesWithSpace)] <- 0

  df <- data.frame(variables = names(variablesWithNA),"NA" = as.vector(variablesWithNA),stringsAsFactors

  df = bind_cols(df,"Empaty" = as.vector(variablesWithEmpaty))
  df = bind_cols(df,"?" = as.vector(variablesWithQuestionMark))
  df = bind_cols(df,"Space" = as.vector(variablesWithSpace))

  df
  #ls <- list(valoresFaltantes = df);
  #ls$totalMuestras <- dim(dataset)[1]
  #ls
}
```

```
# Verificar si existen variables con valores faltantes
hasValoresFaltantes(titanic.data)
```

```
##   variables NA. Empaty ? Space
## 1  Survived   0      0 0     0
## 2    Pclass   0      0 0     0
## 3       Sex   0      0 0     0
## 4       Age 263      0 0     0
## 5     SibSp   0      0 0     0
## 6     Parch   0      0 0     0
## 7      Fare   1      0 0     0
```

```
## 8     Cabin   0   1014 0     0
## 9  Embarked   0      0 2     0
```

Al observar el resultado del análisis anterior, podemos identificar que para las variables Age y Fare presenta valores faltantes (NA). Para la variable Cabin se identifica que presenta una cantidad alta de valores faltantes en el formato vacío (""). y para la variable Embarked se identifica valores faltantes en el formato "?".

Llegados a este punto debemos decidir cómo manejar estos registros que contienen valores desconocidos:

Para el atributo **Embarked** realizamos un análisis de proporción de valores faltantes y lo actualizaremos en función del valor mas frecuente. Existen 2 casos con valor faltante con formato "?", con una proporción del 0.15 %, el valor más frecuentes es "Southampton" con una proporción del 56.98 % .

```
arrange(data.frame(round(prop.table(table(titanic.data$Embarked)),4)*100),-Freq)
```

```
##          Var1  Freq
## 1 Southampton 69.82
## 2   Cherbourg 20.63
## 3  Queenstown  9.40
## 4           ?  0.15
```

```
# actualizamos los valores faltantes con el valor más frecuente
titanic.data$Embarked[titanic.data$Embarked=="?"] <- "Southampton"
titanic.data$Embarked <- droplevels(titanic.data$Embarked) #Eliminamos los niveles no utilizados (?)
```

Para el atributo **Cabin** realizamos un análisis de proporción de valores faltantes. Existen 1014 casos con valor faltante con formato vacío (""), con una proporción del 77.46 %, esto corresponde a más de la mitad de las observaciones. Si intentamos completar los valores faltantes, por alguna de las técnicas de imputación de valores perdidos, debido a la alta cantidad de valores faltantes en este atributo, nos puede generar sesgos en los datos de este atributo. De acuerdo a esto, se decide eliminar el atributo **Cabin** del dataset en estudio.

```
data.frame(Total=sort(colSums(titanic.data == ""), decreasing = TRUE),
           Porcentaje = sort(round(colMeans(titanic.data == "")*100, digits = 2), decreasing = TRUE))["
```

```
##       Total Porcentaje
## Cabin  1014      77.46
```

```
# Eliminamos la variable Cabin
titanic.data <- titanic.data[, !(names(titanic.data) %in% c("Cabin"))]
```

Como podemos observar las variables **SibSp**, **Parch** y **Fare**, presenta datos con valores igual a cero, pero para las variables **SibSp**, **Parch** este valor cero significa que no tienen familiares abordo, de acuerdo a esto el valor cero tiene significado para los datos, y no serán gestionados.

Para la variable **Fare** los valores ceros podria significar un error de datos faltantes, ya que tienen un numero de ticket asignado, o tambien podriamos decir que este cero equivale a que estos ticket fueron entregados por un premio. Para esta actividad asumiremos que es un error y lo consideraremos como valores faltantes.

Calculamos la proporciona de valores ceros en la variable **Fare**, y los remplazamos por el formato de valor faltante (NA), para luego predecir estos valores con el método kNN. Existen 17 casos con valor faltante con formato vacío (0), con una proporción del 1.3 %.

```
# Proporcion en
data.frame(Var = c("Fare"),
           Total = length(titanic.data$Fare[titanic.data$Fare == 0 & !is.na(titanic.data$Fare)]),
           Porcentaje = round((length(titanic.data$Fare[titanic.data$Fare == 0 & !is.na(titanic.data$Fa
```

```
##    Var Total Porcentaje
## 1 Fare    17        1.3
```

```
titanic.data$Fare [titanic.data$Fare == 0 & !is.na(titanic.data$Fare)]<- NA

str(titanic.data)
```

```
## 'data.frame':    1309 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "No","Si": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "Upper","Middle",..: 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp   : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "Cherbourg","Queenstown",..: 3 1 3 3 3 2 3 3 3 1 ...
```

Para los atributo **Fare** y **Age** realizamos un análisis de proporción de valores faltantes. Para el caso del atributo **Fare**, existe 18 caso con valor faltante con formato vacío (NA), con una proporción del 1.38 %; Y para el atributo **Age**, existe 263 casos con valores faltantes con formato vacío (NA), con una proporción del 20.09 %; Debido a que los datos presente en esta variable están un poco dispersos, utilizaremos métodos probalísticos para predecir los valores faltantes.

```
#library(VIM)
if(!require(VIM)){
    #install.packages('VIM', repos='http://cran.us.r-project.org')
    library(VIM)
}
data.frame(Total=sort(colSums(is.na(titanic.data)), decreasing = TRUE),Porcentaje = sort(round(colMeans
```

```
##      Total Porcentaje
## Fare    18       1.38
## Age    263      20.09
```

```
# Para predecir los valores faltantes utilizaremos el metodo kNN
titanic.data.imp <- kNN(titanic.data)

# Imputamos los valores faltantes
titanic.data$Age <- titanic.data.imp$Age # Age
titanic.data$Fare <- titanic.data.imp$Fare #Fare
rm(titanic.data.imp)
```

```
# Verificar si existen variables con valores faltantes
hasValoresFaltantes(titanic.data)
```

```
##   variables NA. Empaty ? Space
## 1  Survived   0      0 0     0
## 2    Pclass   0      0 0     0
## 3       Sex   0      0 0     0
## 4       Age   0      0 0     0
## 5     SibSp   0      0 0     0
## 6     Parch   0      0 0     0
## 7      Fare   0      0 0     0
## 8  Embarked   0      0 0     0
```

### 2.3.3. Identificación y tratamiento de valores extremos

Los valores extremos (outliers) son aquellos datos que se encuentran muy alejados de la distribución normal de una variable o población. Con este análisis queremos identificar si el dataset contiene observaciones que

están alejadas de su distribución normal, con el fin de evitar que estos valores puedan afectar de forma adversa los resultados de los análisis posteriores, al incrementar el error en la varianza de los datos y sesgar significativamente los cálculos y estimaciones.

Para identificar estos valores en el dataset, realizaremos un análisis por cuartiles, para las variables **Age** y **Fare**. Debido a que el resto de variables pueden ser de tipo categóricas o texto no las incluiremos en este análisis.

Realizaremos un análisis de valores extremos para la variable numérica **Age**, realizando un análisis por quartiles:

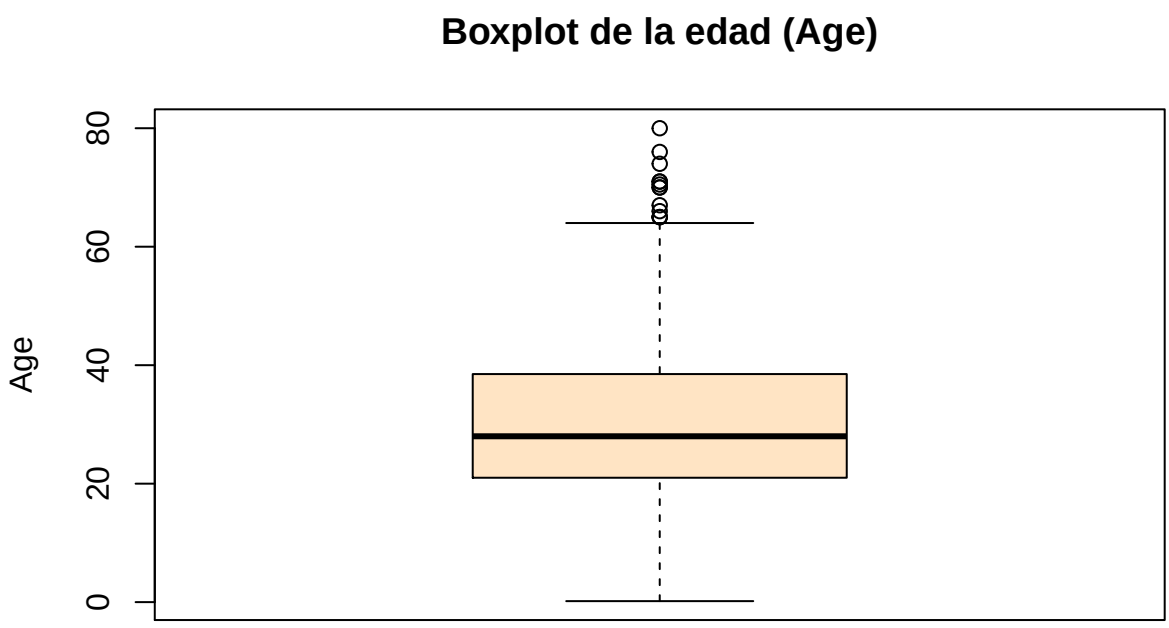
```
# generar los quartiles que representan la distribución del conjunto de datos
summary(titanic.data$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.17   21.00   28.00   29.94   38.50   80.00
```

```
## Calculamos la relación inter quartil (IQR), Q3 - Q1 = IQR()
print(paste("Relación inter quartil (IQR): ",IQR(titanic.data$Age)),quote = FALSE)
```

```
## [1] Relación inter quartil (IQR):  17.5
```

```
# Grafico de boxplot
gf.boxplot <- boxplot(titanic.data$Age, main="Boxplot de la edad (Age)",
ylab="Age",col = "bisque")
```




Al inspeccionar las estadísticas arrojadas, para la variable **Age**, el valor mínimo es 0.17 y el Máximo es 80. Si analisamos la diferencia entre Q1 y el Mínimo es de 20.83, y la diferencia entre Q3 y el Máximo es 42; cómo podemos ver la diferencia de Q3 y el máximo es mayor que la diferencias entre Q1 y el mínimo. Estos nos indican que el 25 % de los valores superiores es tan más dispersos, que el 75 % restante.

Al analizar el grafico de diagrama de cajas (Boxplot), se observa que no hay valores atípicos en el extremo inferior, y por eso el bigote inferior se extiende hasta el valor mínimo, 0.17. En cambio en el extremo superior vemos varios valores atípicos, representados por unos círculos sobre el bigote superior.

Para detectar los valores atípicos, los bigotes se extendieron hasta un $Mínimo = Q1 - 1{,}5 * IQR$, por debajo de Q1 y hasta un $Máximo = Q3 + 1{,}5 * IQR$, por encima de Q3. Donde **IQR = 17**, **Q1 = 21** y **Q3 = 38**; Entonces el $Mínimo = 21 \check{} 1{,}5 * 17 = -4{,}5$, donde todos los valores menores a este valor son considerados atípicos, en nuestro caso como no hay valores menores que este, por eso el bigote se extiende hasta el mínimo

valor de la variable; Los valores mayores al $Máximo = 38 + 1{,}5 * 17 = 63{,}5$ serán considerados atípicos, que son los valores representados en el grafico por los puntos negros.

Considerando lo anterior, a continuación se muestran los valores atípicos para la variable **Age**. **Donde Age > 63.5**:

```
#Valores extremos encontrados en la variable Age donde Age > 63.5
sort(gf.boxplot$out, decreasing = FALSE)
```

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

No obstante, si revisamos los anteriores datos, las edades de los pasajeros comprendidas entre 64 y 80, son valores que perfectamente pueden darse. Es por ello que el manejo de estos valores extremos consistirá en simplemente dejarlos como actualmente están recogidos.

Realizaremos un análisis de valores extremos para la variable numérica **Fare**, realizando un análisis por quartiles:

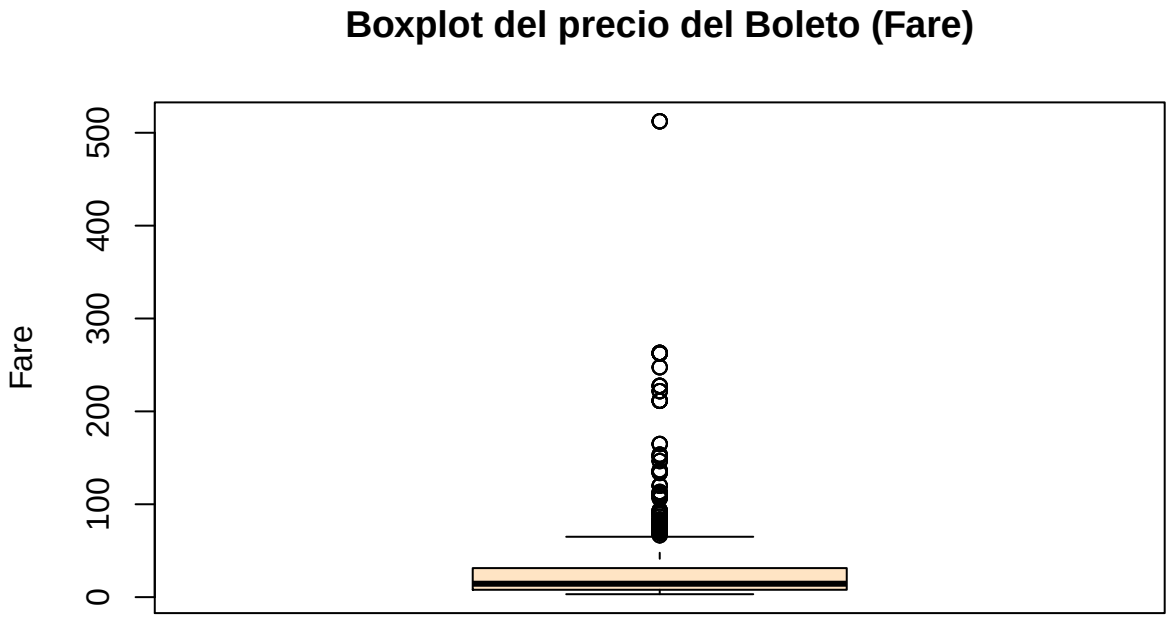
```
# generar los quartiles que representan la distribución del conjunto de datos
summary(titanic.data$Fare)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.171   7.925  14.458  33.484  31.275 512.329
```

```
## Calculamos la relación inter quartil (IQR), Q3 - Q1 = IQR()
print(paste("Relación inter quartil (IQR): ",IQR(titanic.data$Fare)),quote = FALSE)
```

```
## [1] Relación inter quartil (IQR):  23.35
```

```
# Grafico de boxplot
gf.boxplot <- boxplot(titanic.data$Fare, main="Boxplot del precio del Boleto (Fare)",
ylab="Fare",col = "bisque")
```




Al inspeccionar las estadísticas arrojadas, para la variable **Fare**, el valor mínimo es 3.17 y el Máximo es 512.33. Si analisamos la diferencia entre Q1 y el Mínimo es de 4.75, y la diferencia entre Q3 y el Máximo es 481.05; cómo podemos ver la diferencia de Q3 y el máximo es mayor que la diferencias entre Q1 y el mínimo. Estos nos indican que el 25 % de los valores superiores es tan más dispersos, que el 75 % restante.

Al analizar el grafico de diagrama de cajas (Boxplot), se observa que no hay valores atípicos en el extremo

inferior, y por eso el bigote inferior se extiende hasta el valor mínimo, 3.17. En cambio en el extremo superior vemos varios valores atípicos, representados por unos círculos sobre el bigote superior.

Para detectar los valores atípicos, los bigotes se extendieron hasta un $Mínimo = Q1 - 1{,}5 * IQR$, por debajo de Q1 y hasta un $Máximo = Q3 + 1{,}5 * IQR$, por encima de Q3. Donde **IQR = 23.35**, **Q1 = 7.93** y **Q3 = 31.28**; Entonces el $Mínimo = 7{,}93\tilde{}1{,}5 * 23{,}35 = -27{,}01$, donde todos los valores menores a este valor son considerados atípicos, en nuestro caso como no hay valores menores que este, por eso el bigote se extiende hasta el mínimo valor de la variable; Los valores mayores al $Máximo = 31{,}28 + 1{,}5 * 23{,}35 = 66{,}31$ serán considerados atípicos, que son los valores representados en el grafico por los puntos negros.

Considerando lo anterior, a continuación se muestran los valores atípicos para la variable **Fare**. **Donde Fare > 66.31**:

```
#Valores extremos encontrados en la variable Fare donde Fare > 66.31
sort(gf.boxplot$out, decreasing = FALSE)
```

```
##   [1]  66.6000  66.6000  69.3000  69.3000  69.5500  69.5500  69.5500
##   [8]  69.5500  69.5500  69.5500  69.5500  69.5500  69.5500  69.5500
##  [15]  69.5500  71.0000  71.0000  71.2833  71.2833  73.5000  73.5000
##  [22]  73.5000  73.5000  73.5000  73.5000  73.5000  75.2417  75.2417
##  [29]  75.2500  75.2500  76.2917  76.2917  76.7292  76.7292  76.7292
##  [36]  77.2875  77.2875  77.9583  77.9583  77.9583  78.2667  78.2667
##  [43]  78.8500  78.8500  78.8500  79.2000  79.2000  79.2000  79.2000
##  [50]  79.2000  79.2000  79.6500  79.6500  79.6500  80.0000  80.0000
##  [57]  81.8583  81.8583  81.8583  82.1708  82.1708  82.2667  82.2667
##  [64]  83.1583  83.1583  83.1583  83.1583  83.1583  83.1583  83.4750
##  [71]  83.4750  86.5000  86.5000  86.5000  89.1042  89.1042  90.0000
##  [78]  90.0000  90.0000  90.0000  90.0000  91.0792  91.0792  93.5000
##  [85]  93.5000  93.5000  93.5000 106.4250 106.4250 106.4250 108.9000
##  [92] 108.9000 108.9000 110.8833 110.8833 110.8833 110.8833 113.2750
##  [99] 113.2750 113.2750 120.0000 120.0000 120.0000 120.0000 133.6500
## [106] 133.6500 134.5000 134.5000 134.5000 134.5000 134.5000 135.6333
## [113] 135.6333 135.6333 135.6333 136.7792 136.7792 146.5208 146.5208
## [120] 146.5208 151.5500 151.5500 151.5500 151.5500 151.5500 151.5500
## [127] 153.4625 153.4625 153.4625 164.8667 164.8667 164.8667 164.8667
## [134] 211.3375 211.3375 211.3375 211.3375 211.5000 211.5000 211.5000
## [141] 211.5000 211.5000 221.7792 221.7792 221.7792 221.7792 227.5250
## [148] 227.5250 227.5250 227.5250 227.5250 247.5208 247.5208 247.5208
## [155] 262.3750 262.3750 262.3750 262.3750 262.3750 262.3750 262.3750
## [162] 263.0000 263.0000 263.0000 263.0000 263.0000 263.0000 512.3292
## [169] 512.3292 512.3292 512.3292
```

No obstante, si revisamos los anteriores datos, y miramos de forma aleatoria los precios de los Ticket podemos ver que los precios mas altos corresponde a los pasajeros de clase alta (Pclass = "Upper"), y son valores que perfectamente pueden darse. Es por ello que el manejo de estos valores extremos consistirá en simplemente dejarlos como actualmente están recogidos.

```
# Guardamos los datos de nuestro dataset (data.frame)  aun archivo en formato CSV
write.csv(titanic.data, "../datos/titanic.final.csv")
```

## 2.4. Análisis de los datos

### 2.4.1. Selección de los gurpos de datos que se quieren analizar / comparar

Vamos a realizar un análisis preliminar de las variables, para determinar su normalidad y su relación con survived, esto nos ayudará a realizar ejercicios posteriores para tratar de predecir qué pasajeros se salvaron

del titanic y cuales no.

Las variables que vamos a analizar son sex, pclass, age, sibsp,parch,embarked,fare y su relación con survived, que es la variable dependiente que tratamos de predecir. El resto de variables como id, name, ticket o cabin son datos individuales de cada pasajero que no parece que vayan a aportar una información valiosa.

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

Ver apartado 2.5

#### 2.4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

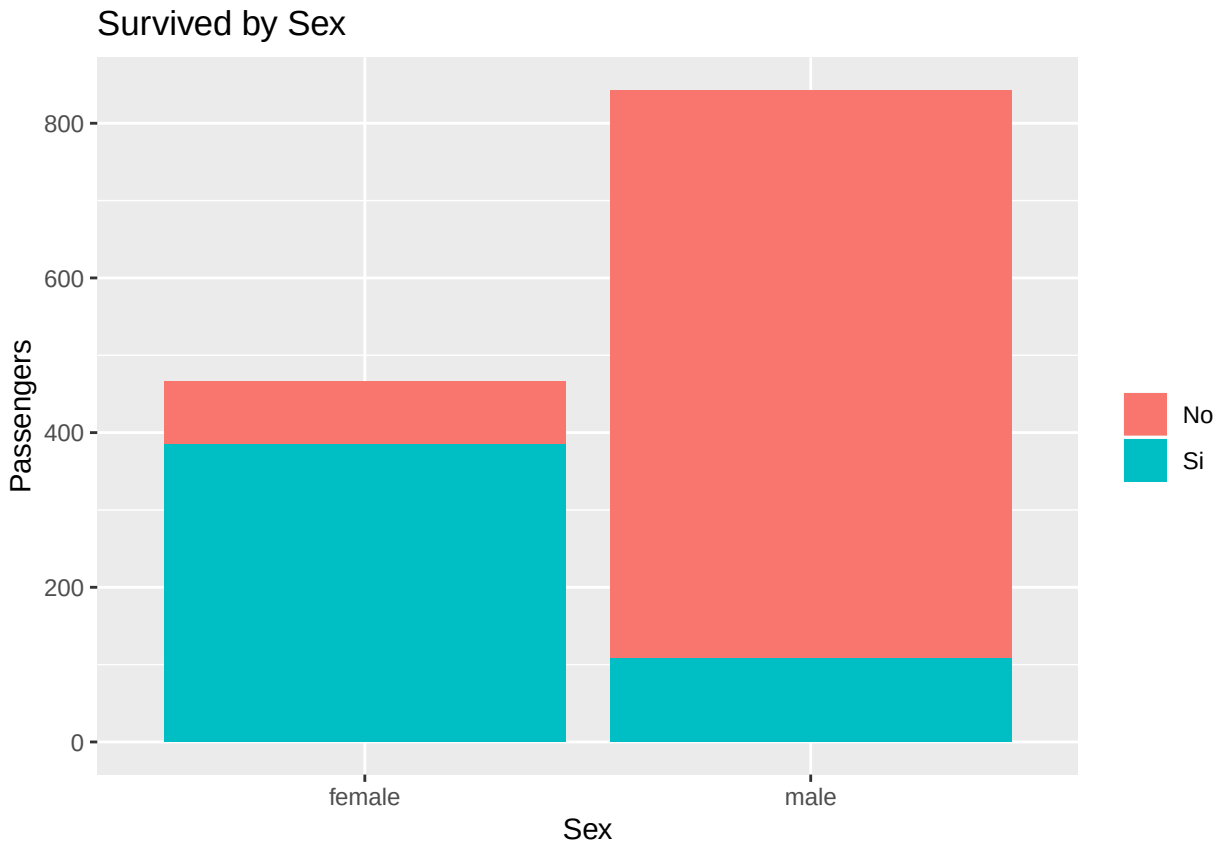
Ver apartado 2.5

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

Por claridad en el informe, en este apartado se compilan los 3 puntos anteriores (apartados 2.4.2 y 2.4.3) ya que se realizan las pruebas estadísticas así como los métodos de análisis pertinentes para determinar qué probabilidades tiene cada persona de haber sobrevivido al hundimiento del Titanic, además se representan las tablas y gráficas pertinentes para una mejor visualización de cada apartado del análisis.

Vamos a comenzar analizando la variable sex con respecto a survived, en este caso podemos realizar una tabla de contingencia. Al ser 2 variables categóricas no tiene sentido analizar la distribución.

```
ggplot(titanic.data,aes(Sex,fill=Survived))+geom_bar() +labs(x="Sex", y="Passengers")+ guides(fill=guid
```




Vemos cómo el porentaje de mujeres supervivientes es mucho mayor que el de hombres, esto lo podemos comprobar con una tabla de contingencia.

```
tblSex<-table(titanic.data$Survived,titanic.data$Sex)
tblSex
```

```
##
##      female male
##   No     81  734
##   Si    385  109
```

Efectivamente vemos cómo el sexo puede ser un factor determinante a la hora de realizar una predicción sobre la supervivencia.

Vamos a ejecutar el test chi-square para asegurar que existen diferencias significativas entre los grupos.

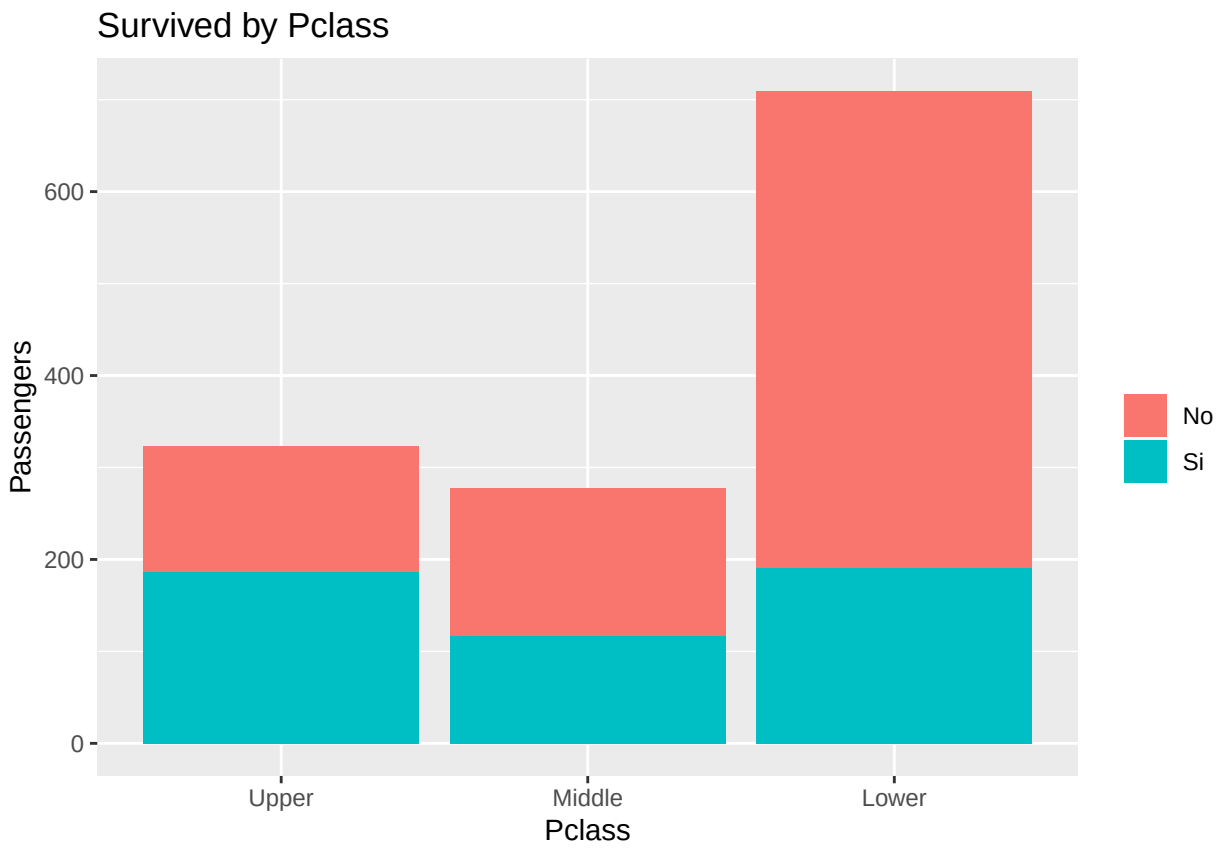
```
chisq.test(tblSex)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tblSex
## X-squared = 617.31, df = 1, p-value < 2.2e-16
```

Efectivamente el test chi-square arroja un p-value bastante pequeño, lo que nos indica que existe una correlación entre estas variables, con lo que el sexo nos puede ayudar a discernir si una persona fue superviviente del titanic o no.

Hacemos el mismo análisis con la variable PClass.

```
library(ggplot2)
ggplot(titanic.data,aes(Pclass,fill=Survived))+geom_bar() +labs(x="Pclass", y="Passengers")+ guides(fil
```




Vemos cómo existe también una diferencia grande en el porcentaje de supervivientes según el nivel económico de los pasajeros.

Vemos la tabla de contingencia.

```
tblClass<-table(titanic.data$Survived,titanic.data$Pclass)
tblClass
```

```
##
##      Upper Middle Lower
##   No   137    160   518
##   Si   186    117   191
```

Ejecutamos el test chi-squared para ver la independencia entre los casos.

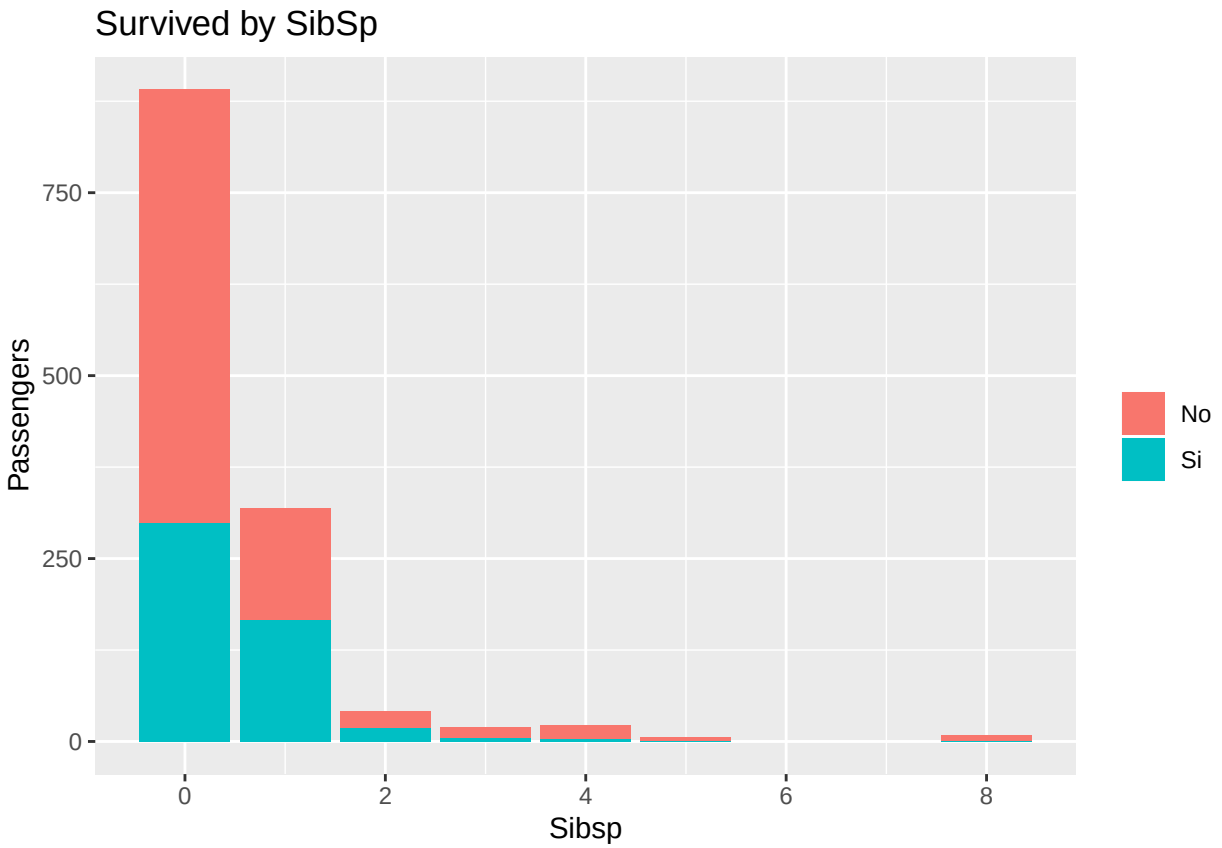
```
 chisq.test(tblClass)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tblClass
## X-squared = 91.724, df = 2, p-value < 2.2e-16
```

En este caso también observamos diferencias significativas en la supervivencia según la variable class, el p-value de nuevo nos indica que debemos rechazar la hipótesis nula de independencia, con lo que vemos cierta correlación entre PClass y Survived.

Analizamos ahora la variable Sibsp de la misma forma.

```
ggplot(titanic.data,aes(SibSp      ,fill=Survived))+geom_bar() +labs(x="Sibsp", y="Passengers")+ guides
```




Atendiendo al gráfico, vemos diferencias significativas en el porcentaje de supervivientes según el número de familiares que están en el barco.

Vamos a realizar al igual que en los casos anteriores la tabla de contingencia y el test chi-square.

```
tblSibsp<-table(titanic.data$Survived,titanic.data$SibSp)
tblSibsp
```

```
##
##        0   1   2   3   4   5   8
##   No 593 153  23  15  18   5   8
##   Si 298 166  19   5   4   1   1
```

Efectivamente se observa cómo el porcentaje de supervivencia varía bastante según el número de familiares en el barco.

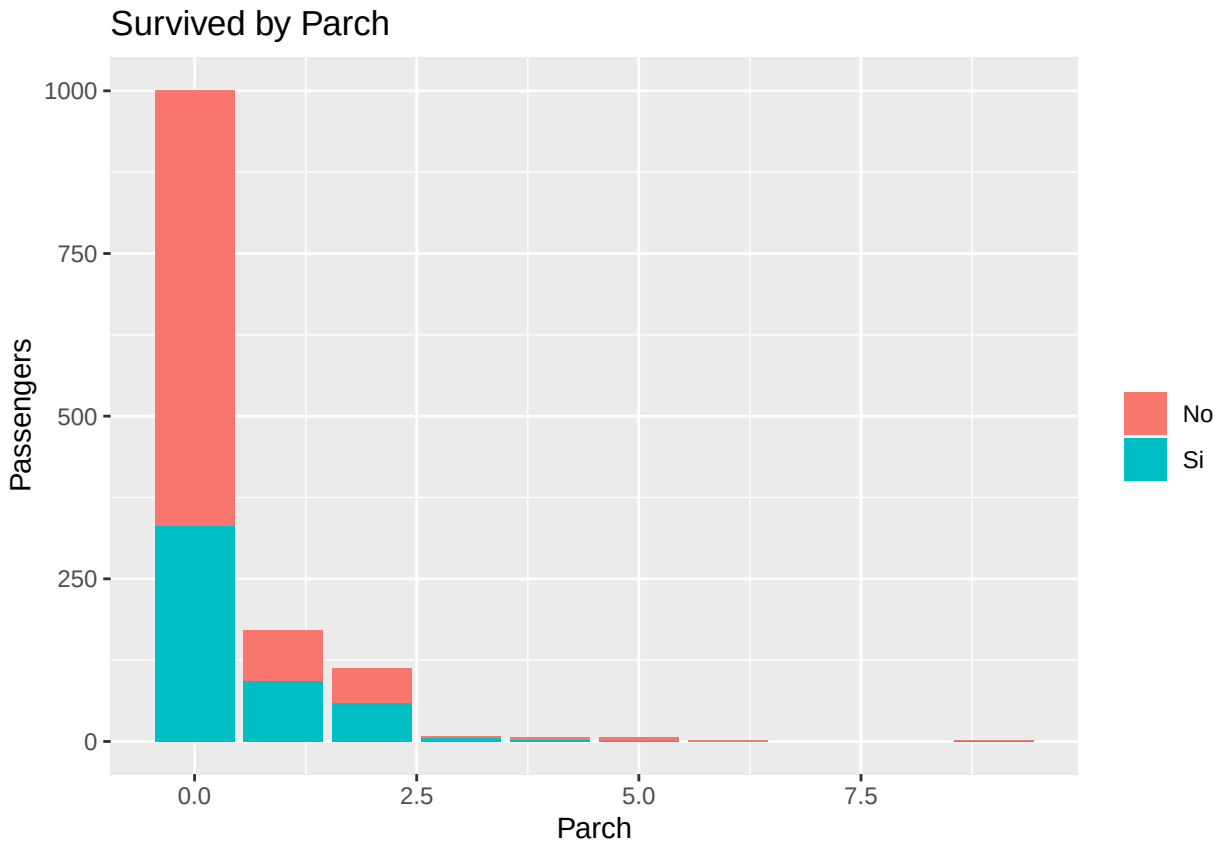
```
chisq.test(tblSibsp)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tblSibsp
## X-squared = 44.565, df = 6, p-value = 5.711e-08
```

De nuevo el test chi-squared arroja cierta correlación entre estas variables, con lo que el número de parientes también puede ser una variable importante a la hora de predecir la supervivencia de una persona.

Repetimos el mismo análisis con la variable Parch (número de padres/hijos)

```
ggplot(titanic.data,aes(Parch,fill=Survived))+geom_bar() +labs(x="Parch", y="Passengers")+ guides(fill=g
```




```
tblParch<-table(titanic.data$Survived,titanic.data$Parch)
tblParch
```

```
##
##        0   1   2   3   4   5   6   9
##   No 670  77  53   3   4   5   2   1
##   Si 332  93  60   5   2   1   0   1
```

Repetimos el test chi-squared

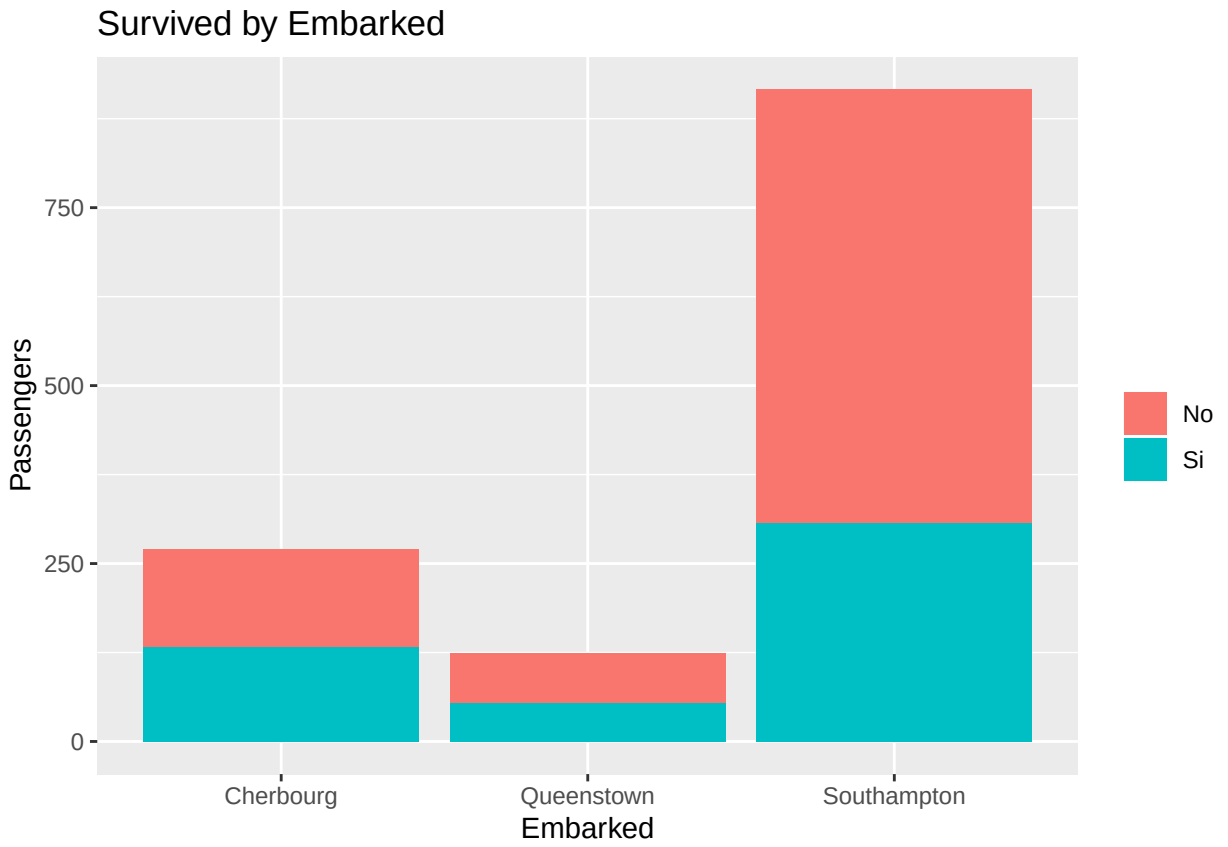
```
chisq.test(tblParch)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tblParch
## X-squared = 45.827, df = 7, p-value = 9.445e-08
```

Estos datos también indican diferencias significativas en el la supervivencia con respecto al número de familiares.

Repetimos el mismo análisis con el puerto de embarque

```
ggplot(titanic.data,aes(Embarked,fill=Survived))+geom_bar() +labs(x="Embarked", y="Passengers")+ guides
```




Según el gráfico también existen diferencias significativas en el porcentaje de supervivientes según el puerto de embarque. Vamos a comprobarlo de nuevo con la tabla de contingencia y el test chi-squared.

```
tblEmbarked<-table(titanic.data$Survived,titanic.data$Embarked)
tblEmbarked
```

```
##
##      Cherbourg Queenstown Southampton
##   No       137         69         609
##   Si       133         54         307
```

```
chisq.test(tblEmbarked)
```

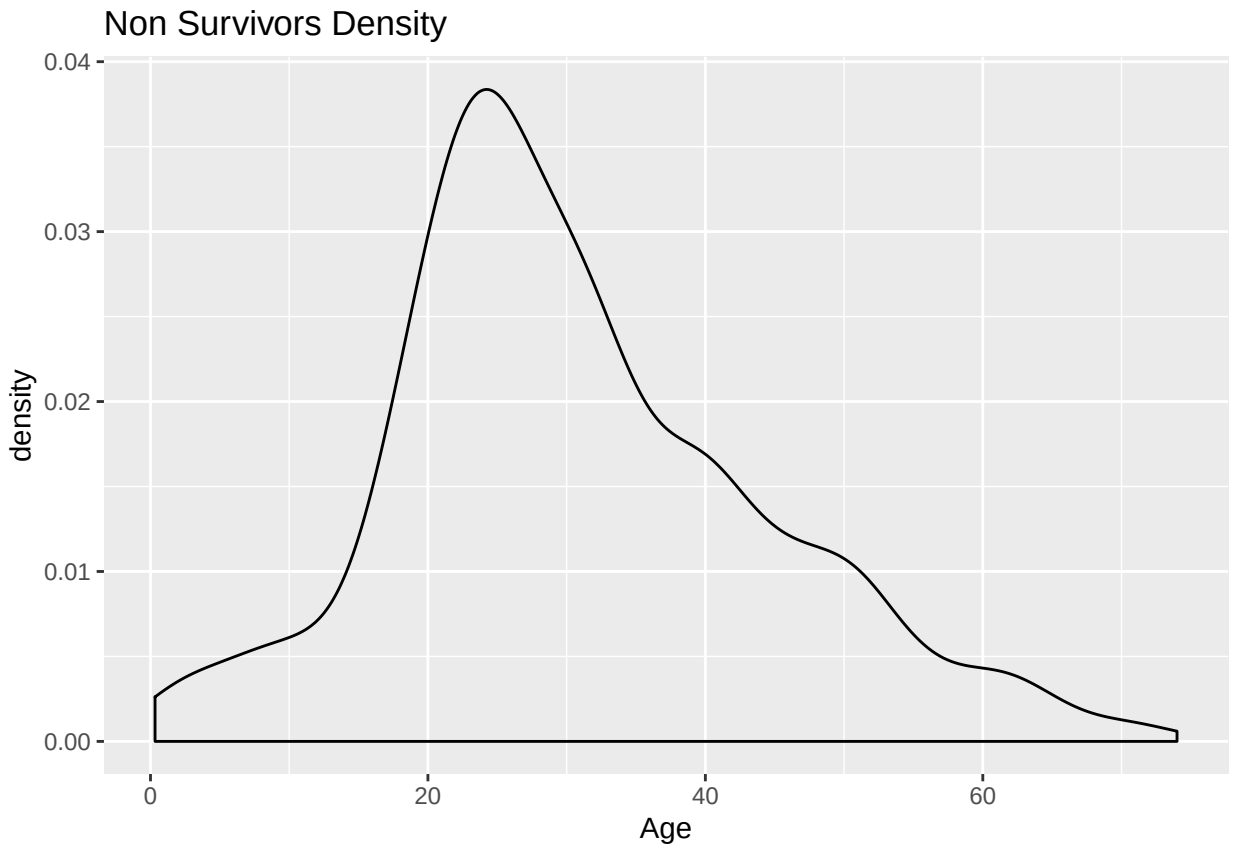
```
##
##  Pearson's Chi-squared test
##
## data:  tblEmbarked
## X-squared = 24.194, df = 2, p-value = 5.577e-06
```

El test chi-squared nos dice de nuevo que sí hay una relación entre el puerto de embarque y la supervivencia.

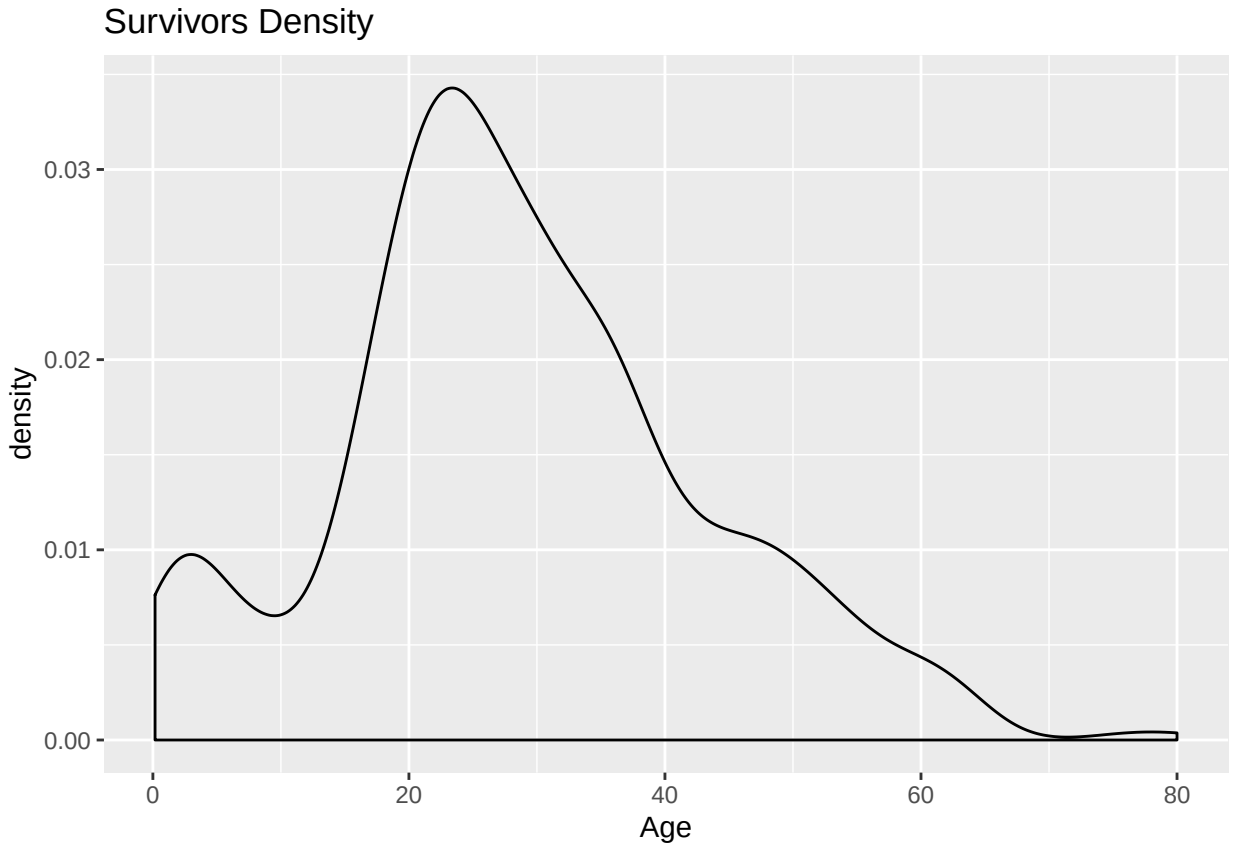
Vamos a analizar la variable edad, en este caso se trata de una variable continua con lo que analizaremos su distribución y veremos si existen diferencias significativas en la media de edad de los supervivientes y los no supervivientes.

Vemos las distribuciones de la edad de los supervivientes y los no supervivientes en las siguientes gráficas.

```
ggplot(titanic.data[titanic.data$Survived=="No",], aes(x=Age)) +
  geom_density()+ ggtitle("Non Survivors Density")
```




```
ggplot(titanic.data[titanic.data$Survived=="Si",], aes(x=Age)) +
  geom_density()+ ggtitle("Survivors Density")
```




Vemos en las gráficas de densidad que ninguna de las 2 sigue una distribución normal. Lo comprobamos con el test de Shapiro Wilk

```
shapiro.test(titanic.data[titanic.data$Survived=="No",]$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanic.data[titanic.data$Survived == "No", ]$Age
## W = 0.97253, p-value = 2.954e-11
```

```
shapiro.test(titanic.data[titanic.data$Survived=="Si",]$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanic.data[titanic.data$Survived == "Si", ]$Age
## W = 0.9796, p-value = 2.101e-06
```

Efectivamente se comprueba que la variable edad no sigue una distribución normal. Vamos a comparar las varianzas entre las edades de ambos grupos.

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 2.0428, df = 1, p-value = 0.1529
```

El p-value del test sugiere que existe homoceasticidad entre ambos grupos, es decir que la varianza entre las edades de los supervivientes y los no supervivientes es parecida.

Dado que existe homoceasticidad entre los grupos y que tenemos un número significativamente alto de muestras, por el teorema del límite central podemos realizar un test t-student para comprobar las medias entre ambos grupos.

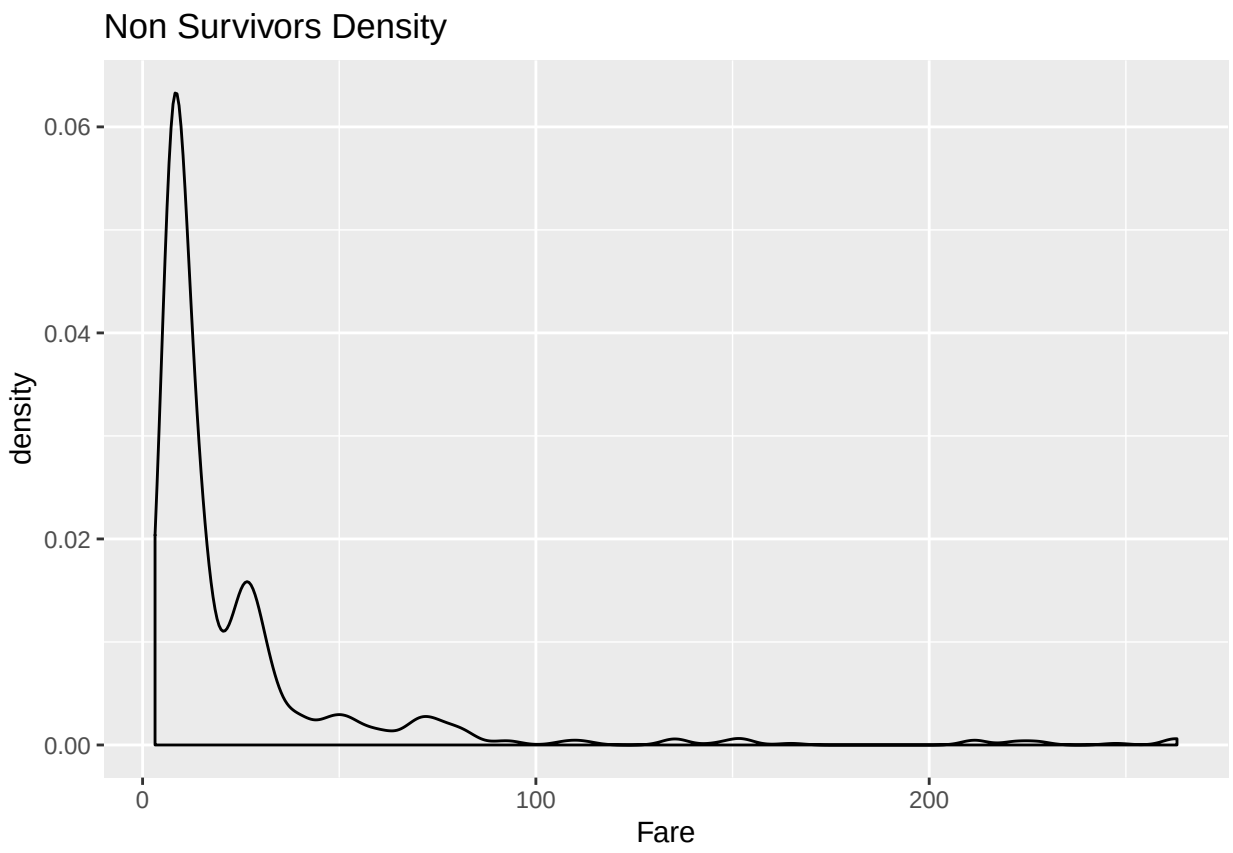
```
t.test(Age ~ Survived,data=titanic.data,alternative="two.sided",var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  Age by Survived
## t = 2.678, df = 1307, p-value = 0.007499
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.5642668 3.6553484
## sample estimates:
## mean in group No mean in group Si
##         30.73179         28.62198
```

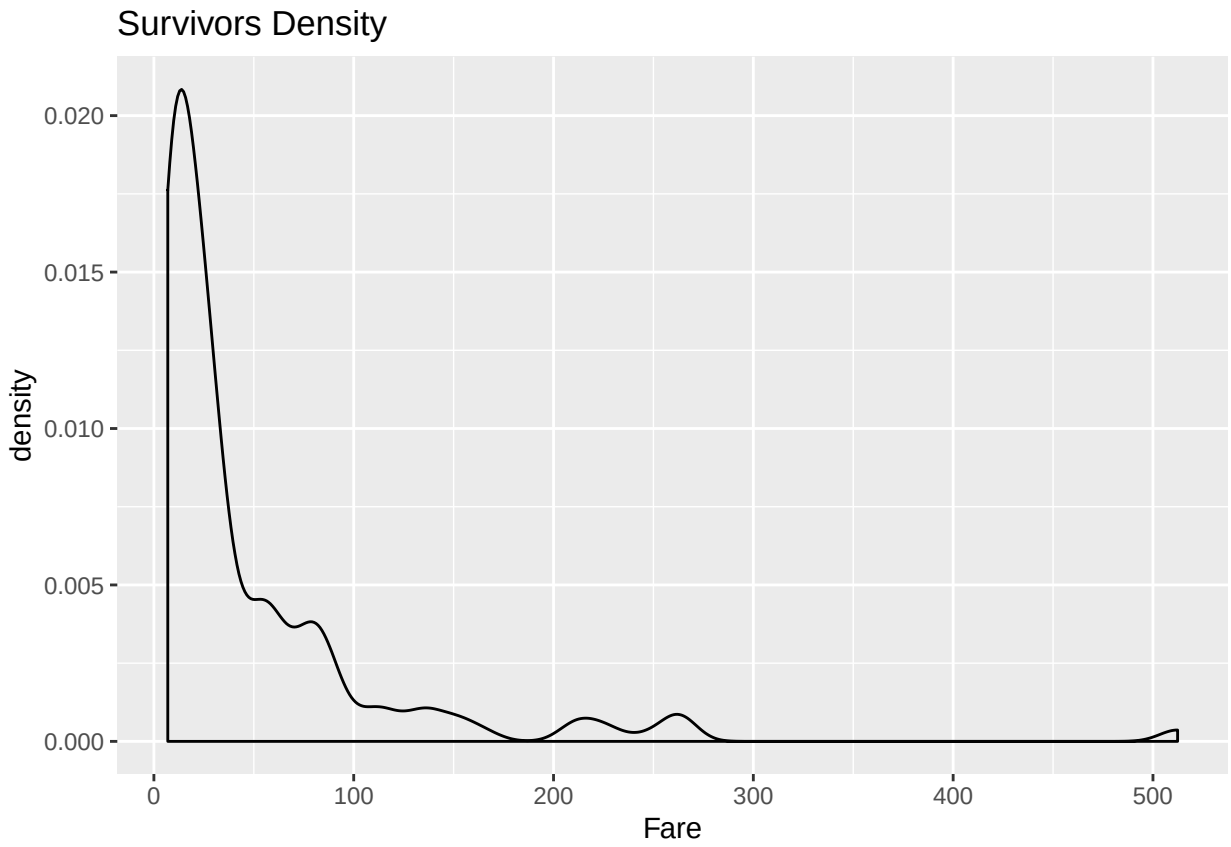
En este caso vemos cómo el p-value es menor a 0.05, con lo que podemos rechazar la hipótesis nula, las medias de edad entre supervivientes y no supervivietnes no son iguales, esto nos dice que la edad puede ser un elemento importante a la hora de clasificar los supervivientes de los no supervivientes.

Vamos a repetir el mismo análisis con la variable Fare frente a Survived.

```
ggplot(titanic.data[titanic.data$Survived=="No",], aes(x=Fare)) +
  geom_density()+ ggtitle("Non Survivors Density")
```



```r
ggplot(titanic.data[titanic.data$Survived=="Si",], aes(x=Fare)) +
  geom_density()+ ggtitle("Survivors Density")
```




Vemos claramente que no hay normalidad en estas variables, vamos a realizar el test de homoceasticidad, como en el caso de la edad.

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by Survived
## Fligner-Killeen:med chi-squared = 128.54, df = 1, p-value <
## 2.2e-16
```

En este caso tampoco se observa homoceasticidad entre ambos grupos, debemos rechazar la hipótesis nula, las varianzas no son parecidas, con lo que para hacer una comparación entre los 2 grupos en este caso debemos usar un test no paramétrico de Wilcoxon ya que son grupos independientes.

```r
wilcox.test(Fare ~ Survived, data = titanic.data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Fare by Survived
## W = 135550, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

No podemos rechazar la hipótesis nula, vemos que sí se observan diferencias significativas entre las tarifas

pagadas por los supervivientes y los no supervivientes.

### 2.5.1. Modelos predictivos.

Utilizaremos las variables anteriores para tratar de construir un modelo predictivo, en todas las variables estudiadas se ha observado cierta relación con la variable survived, con lo que en principio las usaremos en nuestros modelos.

Para realizar los análisis pertientes y tratar de obtener un modelo predictivo, vamos a obtener el dataset de test, seleccionando las filas desde 892 hasta 1309.

```
titanic.test<- titanic.data[892:1309,]
```

Una vez tenemos el dataset de test preparado, vamos a ejecutar una regresión logística sobre los parámetros que hemos elegido, para ver cómo se comporta.

```
logit =glm(formula=Survived ~  Sex+Pclass+Age+SibSp+Parch+Embarked+Fare, data=titanic.data[1:891,],famil
summary(logit)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + Age + SibSp + Parch +
##     Embarked + Fare, family = binomial, data = titanic.data[1:891,
##     ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7074  -0.6257  -0.3911   0.6261   2.7311
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           4.453461   0.492163   9.049  < 2e-16 ***
## Sexmale              -2.687573   0.202275 -13.287  < 2e-16 ***
## PclassMiddle         -1.106648   0.305056  -3.628 0.000286 ***
## PclassLower          -2.419490   0.312792  -7.735 1.03e-14 ***
## Age                  -0.045212   0.007931  -5.700 1.19e-08 ***
## SibSp                -0.375046   0.109763  -3.417 0.000633 ***
## Parch                -0.089018   0.120603  -0.738 0.460447
## EmbarkedQueenstown    0.059824   0.392973   0.152 0.879002
## EmbarkedSouthampton -0.398521   0.240845  -1.655 0.097990 .
## Fare                  0.001503   0.002423   0.620 0.535116
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.7  on 890  degrees of freedom
## Residual deviance:  775.5  on 881  degrees of freedom
## AIC: 795.5
##
## Number of Fisher Scoring iterations: 5
```

Observamos que según el p-value las variables más importantes para diferenciar supervivientes de no supervivientes son Sex, PClass, Age, SibSp.

Realizamos una predicción sobre las mismas columnas de test, esto nos dará la probabilidad de supervivencia,

consideramos supervivientes los que tengan una probabilidad mayor de 0.5

```
prediction<-predict(logit,newdata=titanic.test[,!(colnames(titanic.test) %in% c("Survived"))],type="resp

surv_prediction = ifelse(prediction>0.5,1,0)

table(surv_prediction)
```

```
## surv_prediction
##   0   1
## 263 155
```

Para comparar los resultados realizamos la matriz de confusión con los valores que tenemos en test.

```
conf_Matrix<-table(surv_prediction,titanic.test$Survived)
conf_Matrix
```

```
##
## surv_prediction  No  Si
##               0 250  13
##               1  16 139
```

```
porcentaje_correcto<-100 * sum(diag(conf_Matrix)) / sum(conf_Matrix)
porcentaje_correcto
```

```
## [1] 93.0622
```

Observamos cómo por este método se han clasificado correctamente 250 no supervivientes y 139 supervivientes.

Esto hace una fiabilidad de la predicción del 93.06 %

Vamos a intentar clasificar y predecir los supervivientes mediante un árbol de decisión. Utilizamos la función rpart.

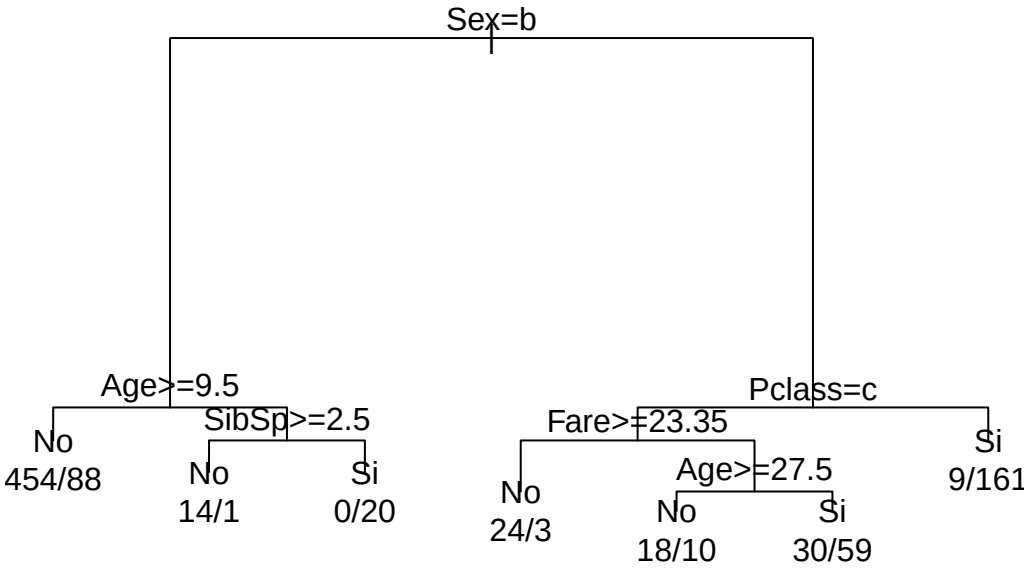
```
library(rpart)
model_cart<-rpart(Survived~Sex+Pclass+Age+SibSp+Parch+Embarked+Fare,data=titanic.data[1:891,],method="cl
model_cart
```

```
## n= 891
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 891 342 No (0.61616162 0.38383838)
##    2) Sex=male 577 109 No (0.81109185 0.18890815)
##      4) Age>=9.5 542  88 No (0.83763838 0.16236162) *
##      5) Age< 9.5 35  14 Si (0.40000000 0.60000000)
##       10) SibSp>=2.5 15   1 No (0.93333333 0.06666667) *
##       11) SibSp< 2.5 20   0 Si (0.00000000 1.00000000) *
##    3) Sex=female 314  81 Si (0.25796178 0.74203822)
##      6) Pclass=Lower 144  72 No (0.50000000 0.50000000)
##       12) Fare>=23.35 27   3 No (0.88888889 0.11111111) *
##       13) Fare< 23.35 117  48 Si (0.41025641 0.58974359)
##         26) Age>=27.5 28  10 No (0.64285714 0.35714286) *
##         27) Age< 27.5 89  30 Si (0.33707865 0.66292135) *
##      7) Pclass=Upper,Middle 170   9 Si (0.05294118 0.94705882) *
```

Vamos a mostrar el árbol

```
par(xpd = NA)

plot(model_cart)
text(model_cart,use.n=TRUE)
```




Vamos a validar este modelo con el set de datos de test:

```
predicted.classes <- predict( model_cart, titanic.test[,!(colnames(titanic.test) %in% c("Survived"))], t
```

Vamos a validar el resultado de la predicción con el árbol de decisión

```
conf_Matrix<-table(predicted.classes,titanic.test$Survived)
conf_Matrix
```

```
##
## predicted.classes  No  Si
##                No 257  23
##                Si   9 129
```

```
porcentaje_correcto<-100 * sum(diag(conf_Matrix)) / sum(conf_Matrix)
porcentaje_correcto
```

```
## [1] 92.3445
```

Observamos que con este método disminuyo el porcentaje de acierto en la predicción a un 92.34 %. El mayor problema son los falsos positivos y los falsos negativos, es decir, que el algoritmo ha clasificado como supervivientes 13 casos que en realidad son no supervivientes y clasifico como no supervivientes 16 casos que en realidad son supervivientes.

Para ajustar esto podemos asignar penalizaciones a los falsos negativos y falsos positivos, penalizando más lo que nos interesa.Vamos a repetir el método ajustando las penalizaciones con el parámetro loss.

```
library(rpart)
model_cart<-rpart(Survived~Sex+Pclass+Age+SibSp+Parch+Embarked+Fare,data=titanic.data[1:891,],method="cl
model_cart
```

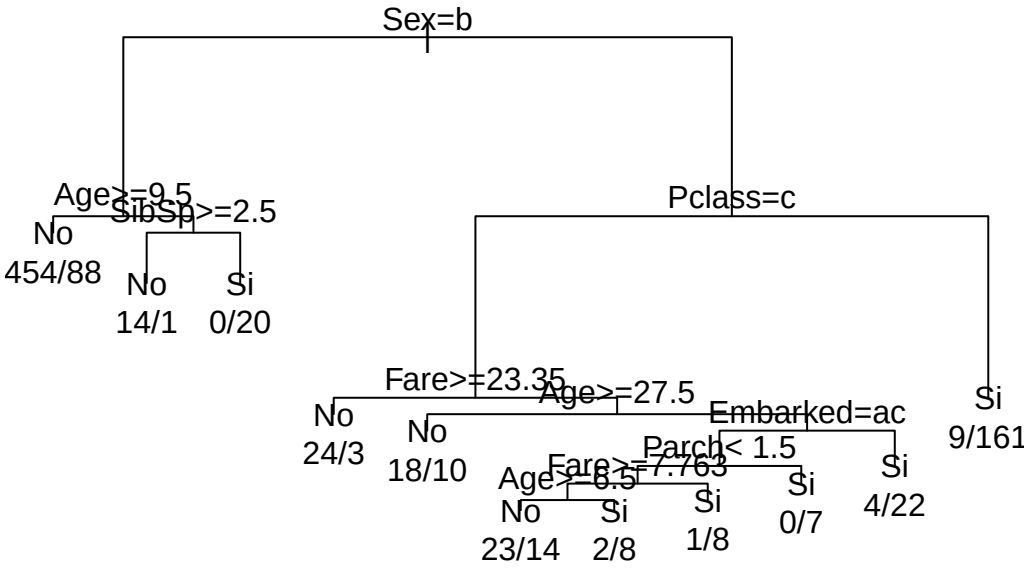
```
## n= 891
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
```

```
##
##   1) root 891 171.0 No (0.61616162 0.38383838)
##     2) Sex=male 577  54.5 No (0.81109185 0.18890815)
##       4) Age>=9.5 542  44.0 No (0.83763838 0.16236162) *
##       5) Age< 9.5 35  10.5 No (0.40000000 0.60000000)
##        10) SibSp>=2.5 15   0.5 No (0.93333333 0.06666667) *
##        11) SibSp< 2.5 20   0.0 Si (0.00000000 1.00000000) *
##     3) Sex=female 314  81.0 Si (0.25796178 0.74203822)
##       6) Pclass=Lower 144  36.0 No (0.50000000 0.50000000)
##        12) Fare>=23.35 27   1.5 No (0.88888889 0.11111111) *
##        13) Fare< 23.35 117  34.5 No (0.41025641 0.58974359)
##          26) Age>=27.5 28   5.0 No (0.64285714 0.35714286) *
##          27) Age< 27.5 89  29.5 No (0.33707865 0.66292135)
##            54) Embarked=Cherbourg,Southampton 63  18.5 No (0.41269841 0.58730159)
##             108) Parch< 1.5 56  15.0 No (0.46428571 0.53571429)
##               216) Fare>=7.7625 47  11.0 No (0.53191489 0.46808511)
##                 432) Age>=6.5 37   7.0 No (0.62162162 0.37837838) *
##                 433) Age< 6.5 10   2.0 Si (0.20000000 0.80000000) *
##               217) Fare< 7.7625 9   1.0 Si (0.11111111 0.88888889) *
##             109) Parch>=1.5 7   0.0 Si (0.00000000 1.00000000) *
##            55) Embarked=Queenstown 26   4.0 Si (0.15384615 0.84615385) *
##       7) Pclass=Upper,Middle 170   9.0 Si (0.05294118 0.94705882) *
```

Mostramos el nuevo árbol.

```
par(xpd = NA)

plot(model_cart)
text(model_cart,use.n=TRUE)
```




Realizamos la predicción.

```
predicted.classes <- predict( model_cart, titanic.test[,!(colnames(titanic.test) %in% c("Survived"))],
```

Hallamos la matriz de confusión.

```
conf_Matrix<-table(predicted.classes,titanic.test$Survived)
conf_Matrix
```

```
##
## predicted.classes  No  Si
##                No 257  46
##                Si   9 106
```

```
porcentaje_correcto<-100 * sum(diag(conf_Matrix)) / sum(conf_Matrix)
porcentaje_correcto
```

```
## [1] 86.84211
```

Después de realizar el ajuste con los pesos, vemos cómo obtenemos un árbol más sencillo y efectivo ya que llegamos a un 97.28 % de clasificación correcta.

Por último vamos a intentar clasificar mediante un modelo de clústering por el algoritmo mclust, que es un algoritmo basado en modelos. En este caso usa un modelo de mezclas gaussianas, que estima la probabilidad de que un dato pertenezca a cada una de las distribuciones, definidas por su media y su varianza. Para asignar los datos a las distribuciones usa el algoritmo de Esperanza-Maximización.

Seleccionamos las columnas de los anteriores modelos e indicamos que sólo queremos 2 grupos.

```
train<-titanic.data[1:891,!(colnames(titanic.test) %in% c("Survived"))]

levels(train$Sex)<-c(0,1)

train$Sex<-as.numeric(as.character(train$Sex))

#train<-train[train$Embarked!="",]
#levels(train$Embarked)<-droplevels(train$Embarked)
levels(train$Embarked)<-c(0,1,2)
train$Embarked<-as.numeric(as.character(train$Embarked))
levels(train$Pclass)<-c(0,1,2)
train$Pclass<-as.numeric(as.character(train$Pclass))


library(mclust)
fit <- Mclust(train,G=1:2)
summary(fit)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##       -9815.732 891 65 -20072.97 -20073.38
##
## Clustering table:
##   1   2
## 709 182
```

Vamos a realizar la predicción sobre test, para ello debemos transformar las columnas a numérico, de la misma manera que en train.

```
#install.packages("clue")
mtest<-titanic.test
```

```
levels(mtest$Sex)<-c(0,1)

mtest$Sex<-as.numeric(as.character(mtest$Sex))

levels(mtest$Embarked)<-c(0,1,2)
mtest$Embarked<-as.numeric(as.character(mtest$Embarked))

levels(mtest$Pclass)<-c(0,1,2)
mtest$Pclass<-as.numeric(as.character(mtest$Pclass))

mpredict<-predict.Mclust(fit,mtest[,!(colnames(titanic.test) %in% c("Survived"))])
```

Vamos a evaluar el algoritmo de la misma forma que antes, con la matriz de confusión.

```
conf_Matrix<-table(mpredict$classification,mtest$Survived)
conf_Matrix
```

```
##
##      No  Si
##   1 195 152
##   2  71   0
```

```
porcentaje_correcto<-100 * sum(diag(conf_Matrix)) / sum(conf_Matrix)
porcentaje_correcto
```

```
## [1] 46.65072
```

En este caso vemos que el algoritmo se queda en un 59 % de efectividad, bastante menor que los anteriores.

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

Hemos comprobado cómo las variables del dataset que habíamos elegido pueden servir para resolver el problema que es predecir la probabilidad de supervivencia de una persona en el hundimiento del Titanic según sus características.

Primero hemos determinado mediante pruebas estadísticas si existe relación entre las variables del dataset y la variable que indica si una persona es superviviente o no, hemos visto que todas las variables pueden ayudarnos en mayor o menor medida a discernir si una persona fue o no superviviente, aunque unas con más significación que otras.

En cuanto a los modelos predictivos, la regresión logística obtiene buenos resultados con un 90 % de acierto en los datos de test, sin embargo el árbol de decisión, ajustando las penalizaciones adecuadamente obtiene un excelente resultado y acierta un 97 % de los casos. En cuanto a los modelos de clústering, vemos cómo no se ajustan bien para este caso ya que su eficacia se queda en un 60 %.

## 2.7. Tabla de contribuciones al trabajo

| Contribuciones | Firma |
|---|---|
| Investigación previa | RT, DO |
| Redacción de las respuestas | RT, DO |
| Desarrollo código | RT, DO |

# 3. Recursos