M2.851 - Tipología y ciclo de vida de los datos: Practico 2

Autores: Gabriel Álvarez Morgado y Héctor Alejandro Castillo Jeria

Junio 2022

Contents

Introducción

Este documento contiene el desarrollo del Práctico número 2, en el cual, 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.

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

---

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.

Importancia de los análisis

A partir de este conjunto de datos se plantea la problemática de determinar qué variables son las que más influyen en la sobrevivencia o no de los pasajeros del RMS Titanic.

Además, se pretende crear modelos predictivos para emplearlos en el set de datos de test y determinar las probabilidades de que un pasajero pueda o no sobrevivir a la tragedia.

Inicio de Actividad.

Comprensión de los datos.

La muestra con la que trabajaremos corresponde a los datos de pasajeros del **RMS Titanic**, famoso transatlantico que sufre un accidente y se hunde en su viaje inaugural, el día 15 de Abril de 1912, donde fallecen 1502 de sus 2224 pasajeros y tripulantes.

Este juego de datos se encuentra en los archivos **train.csv** y **test.csv** ambos archivos, obtenido desde Kaggle **"Titanic - Machine Learning from Disaster"**

El archivo **train.csv** posee la información de los pasajeros, lo que permitirá entrenar nuestro modelo, para emplear luego el archivo **test.csv** que contiene información de pasajeros, para predecir si los pasajeros de esta muestra sobreviven al hundimiento del Titanic, de acuerdo a ciertos factores que obtendremos en el desarrollo de práctico.

Carga de librerias y fichero de datos.

Instalamos y cargamos las librerías necesarias para desarrollar el práctico.

```
librerias = c("ggplot2", "dplyr", "knitr", "grid", "gridExtra", "ggstatsplot",
              "car","rpart","rpart.plot","caret")

for (lib in librerias){
  if (!require(lib, character.only = TRUE)){
    install.packages(lib)
    library(lib, character.only = TRUE)
  }
}
```

Cargamos los ficheros de datos *train.csv* y *test.csv*.

```
trainData = read.csv('train.csv',stringsAsFactors = FALSE)
testData  = read.csv('test.csv' ,stringsAsFactors = FALSE)
```

Exploración de la base de datos de test

Calcularemos las dimensiones de nuestra base de datos y analizaremos qué tipos de atributos tenemos.

```
dim(testData)
```

```
## [1] 418  11
```

Mediante la función dim(), vemos que tenemos 418 registros (filas) que se corresponden a las reseñas y 11 variables (columnas) que los caracterizan.

Verificamos la estructura del juego de datos principal.

```
str(testData)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Revisamos la descripción de las variables contenidas al fichero y si los tipos de variable se corresponde al que hemos cargado:

Atributo	Descripción
PassengerId	Número Identificador del pasajero
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Name	Nombre
Sex	Sexo
Age	Edad
SibSp	Número de hermanos/cónyuges a bordo
Parch	Número de padres/hijos a bordo
Ticket	Número de Ticket
Fare	Tarifa de pasajero
Cabin	Cabina
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)

Obtendremos estadísticas básicas, para luego trabajar con los atributos que no poseen valores o se encuentran vacíos.

Estadísticas básicas

```
summary(testData)
```

```
##   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
```

Como parte de la exploración de los datos, verificaremos si hay valores perdidos.

```
colSums(is.na(testData) | testData=="")
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          86           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           1         327           0
```

Podemos observar que hay 86 datos faltantes en la variable Age, 327 en el atributo cabin y 1 en Fare.

Las estadísticas obtenidas indican lo siguiente sobre la data:

El atributo **Age**, posee 86 datos vacíos, como este dato es relevante, el atributo no será eliminado y serán imputados los valores faltantes según el promedio de la edad de los pasajeros, dependiendo del sexo, la clase, su estado de sobrevivencia y las variables familiares SibSp y Parch.

El atributo **Fare**, posee 1 dato vacío, se cargará el promedio por defecto.

El atributo **Cabin**, posee 327 datos vacíos, este dato no es relevante y se eliminará el atributo.

El atributo **Name**, este dato no es relevante y se eliminará el atributo.

El atributo **PassengerId**, este dato no es relevante y se eliminará el atributo.

Exploración de la base de datos de train

Calcularemos las dimensiones de nuestra base de datos y analizaremos qué tipos de atributos tenemos.

```
dim(trainData)
```

```
## [1] 891  12
```

Mediante la función dim(). Obtenemos que disponemos de 891 registros (filas) y 12 variables (columnas).

Verificamos la estructura del juego de datos principal.

```
str(trainData)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
```

```
##  $ Age         : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp       : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch       : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket      : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare        : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin       : chr  "" "C85" "" "C123" ...
##  $ Embarked    : chr  "S" "C" "S" "S" ...
```

Revisamos la descripción de las variables contenidas al fichero y si los tipos de variable se corresponde al que hemos cargado:

Atributo	Descripción
PassengerId	Número Identificador del pasajero
Survived	Sobreviviente (0=No; 1=Si)
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Name	Nombre
Sex	Sexo
Age	Edad
SibSp	Número de hermanos/cónyuges a bordo
Parch	Número de padres/hijos a bordo
Ticket	Número de Ticket
Fare	Tarifa de pasajero
Cabin	Cabina
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)

Obtendremos estadísticas básicas, para luego trabajar con los atributos que no poseen valores o se encuentran vacíos.

Estadísticas básicas

```
summary(trainData)
```

```
##   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
##
```

Como parte de la exploración de los datos, verificaremos si hay valores perdidos.

```
colSums(is.na(trainData) | trainData=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

Podemos observar que hay 177 datos faltantes en la variable Age, 687 en el atributo cabin y 2 en Embarked.

> El atributo **Age**, posee 177 datos vacíos, como este dato es relevante, el atributo no será eliminado y serán imputados los valores faltantes según el promedio de la edad de los pasajeros, dependiendo del sexo, la clase, su estado de sobrevivencia y las variables familiares SibSp y Parch.
>
> El atributo **Cabin**, posee 687 datos vacíos, este dato no es relevante y se eliminará el atributo.
>
> El atributo **Embarked**, posee 2 datos vacíos, se cargará un valor por defecto correspondiente a la moda de la variable.
>
> El atributo **Name**, este dato no es relevante y se eliminará el atributo.
>
> El atributo **PassengerId**, este dato no es relevante y se eliminará el atributo.

Preparación de los datos.

En este paso realizaremos la normalización de algunos atributos, la clusterización de otros, la eliminación de datos no relevantes, la creación de nuevos atributos, basicamente en este punto tomamos el set de datos, para luego aplicar tecnicas de normalización, completar atributos inexistentes, agregar valores por defecto, incorporar un atributo empleando formulas estadisticas como la media. Como ejemplo. En este punto, a los atributos de tipo texto, le cargaremos el valor por defecto “desconocido”, cuando el atributo es nulo o vacío.

Reemplazo de valores nulos o vacíos.

Reemplazo de la edad en trainData

```
for(i in seq(nrow(trainData))){
  if(is.na(trainData$Age[i])){
    sim_age = trainData$Age[trainData$Survived==trainData$Survived[i] &
                            trainData$Pclass==trainData$Pclass[i] &
                            trainData$Sex==trainData$Sex[i] &
                            trainData$SibSp==trainData$SibSp[i] &
                            trainData$Parch==trainData$Parch[i]]
    if(is.na(mean(sim_age, na.rm = TRUE))){
      trainData$Age[i] = mean(trainData$Age, na.rm = TRUE)
    } else {
```

```
      trainData$Age[i] = mean(sim_age, na.rm = TRUE)
    }
  }
}
```

Reemplazo de la edad en testData

```
for(i in seq(nrow(testData))){
  if(is.na(testData$Age[i])){
    sim_age = testData$Age[testData$Survived==testData$Survived[i] &
                           testData$Pclass==testData$Pclass[i] &
                           testData$Sex==testData$Sex[i] &
                           testData$SibSp==testData$SibSp[i] &
                           testData$Parch==testData$Parch[i]]
    if(is.na(mean(sim_age, na.rm = TRUE))){
      testData$Age[i] = mean(testData$Age, na.rm = TRUE)
    } else {
      testData$Age[i] = mean(sim_age, na.rm = TRUE)
    }
  }
}
```

Reemplazo de la variable Embarked en trainData y de la variable Fare en testData

```
trainData$Embarked[trainData$Embarked==""] = "S"
testData$Fare[is.na(testData$Fare)] = mean(testData$Fare, na.rm = TRUE)
```

Verificación de corrección de valores vacíos en **train** y **test**.

```
colSums(is.na(testData) | testData=="")
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0           0           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0         327           0
```

```
colSums(is.na(trainData) | trainData=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           0
```

Se visualizan valores faltantes solo en la variable Cabin que será eliminada posteriormente.

Identificación de valores atípicos en trainData

Se observará la presencia de valores extremos o atípicos en las variables numéricas a través del método propuesto por Tukey. Se partirá con la edad:

```
q = quantile(trainData$Age,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC

atip = length(trainData$Age[trainData$Age<LI | trainData$Age>LS])

paste0("Para la variable Age, hay ", atip, " valores atípicos que se encuentran fuera del intervalo (",
```

```
## [1] "Para la variable Age, hay 28 valores atípicos que se encuentran fuera del intervalo (-0.125,58.8
```

Hay 28 datos atípicos en la variable Age, sin embargo se ha decidido no eliminarlos ni trabajarlos como valroes faltantes, puesto que corresponden a edades correctas que posiblemente hayan tenido algunas personas. Como la gran mayoría de los pasajeros eran personas jóvenes, se identifica como valores atípicos aquellas edades más avanzadas.

Ahora pasaremos a las variables familiares SibSP y Parch

```
q = quantile(trainData$SibSp,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC

atip = length(trainData$SibSp[trainData$SibSp<LI | trainData$SibSp>LS])

paste0("Para la variable SibSp, hay ", atip, " valores atípicos que se encuentran fuera del intervalo (
```

```
## [1] "Para la variable SibSp, hay 46 valores atípicos que se encuentran fuera del intervalo (-1.5,2.5)
```

```
q = quantile(trainData$Parch,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC

atip = length(trainData$Parch[trainData$Parch<LI | trainData$Parch>LS])

paste0("Para la variable Parch, hay ", atip, " valores atípicos que se encuentran fuera del intervalo (
```

```
## [1] "Para la variable Parch, hay 213 valores atípicos que se encuentran fuera del intervalo (0,0)"
```

Aquí nuevamente ocurre un fenómeno interesante. Como más del 75% de los pasajeros no tenía hijos a bordo, el método de Tukey identifica como valor atípico aquellos casos que sí tienen hijos. No obstante, los valores que posee este atributo son presumiblemente correctos, por lo que no serán eliminados. Misma política fue aplicada para la variable SibSp.

Finalmente, revisaremos la variable Fare

```
q = quantile(trainData$Fare,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC
```

```
atip = length(trainData$Fare[trainData$Fare<LI | trainData$Fare>LS])

paste0("Para la variable Fare, hay ", atip, " valores atípicos que se encuentran fuera del intervalo ("
```

```
## [1] "Para la variable Fare, hay 116 valores atípicos que se encuentran fuera del intervalo (-26.724,
```

En esta variable se identifican 116 valores atípicos correspondientes a los valores más altos. También se puede presumir que estos datos son identificados como atípicos pues son muy pocos los boletos que tenían un alto precio. SIn embargo no son valores erróneos y se mantendran como están.

Identificación de valores atípicos en testData

Para el conjunto de datos de prueba, ocurren las mismas situaciones que para el conjunto de datos de entrenamiento, por lo que se tomaron las mismas decisiones de mantención de la información entregada en el dataset.

```
q = quantile(testData$Age,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC

atip = length(testData$Age[testData$Age<LI | testData$Age>LS])

paste0("Para la variable Age, hay ", atip, " valores atípicos que se encuentran fuera del intervalo (",
```

```
## [1] "Para la variable Age, hay 36 valores atípicos que se encuentran fuera del intervalo (3.875,54.8
```

```
q = quantile(testData$SibSp,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC

atip = length(testData$SibSp[testData$SibSp<LI | testData$SibSp>LS])

paste0("Para la variable SibSp, hay ", atip, " valores atípicos que se encuentran fuera del intervalo (
```

```
## [1] "Para la variable SibSp, hay 11 valores atípicos que se encuentran fuera del intervalo (-1.5,2.5)
```

```
q = quantile(testData$Parch,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC

atip = length(testData$Parch[testData$Parch<LI | testData$Parch>LS])

paste0("Para la variable Parch, hay ", atip, " valores atípicos que se encuentran fuera del intervalo (
```

```
## [1] "Para la variable Parch, hay 94 valores atípicos que se encuentran fuera del intervalo (0,0)"
```

```
q = quantile(testData$Fare,seq(0.25,0.75,0.25))
RIC =  q[3] - q[1]
LI = q[1] - 1.5*RIC
LS = q[3] + 1.5*RIC

atip = length(testData$Fare[testData$Fare<LI | testData$Fare>LS])

paste0("Para la variable Fare, hay ", atip, " valores atípicos que se encuentran fuera del intervalo ("
```

```
## [1] "Para la variable Fare, hay 55 valores atípicos que se encuentran fuera del intervalo (-27.5105,
```

Eliminación, creación y modificación de atributos.

Se eliminarán las variables que no serán consideradas en los análisis.

```
testData  <- subset( testData, select = -c(PassengerId, Cabin, Name, Ticket ) )
trainData <- subset( trainData, select = -c(PassengerId, Cabin, Name, Ticket ) )
```

Agregaremos un nuevo campo a los datos. Este campo contendrá el valor de la edad discretizada con un método simple de intervalos de igual amplitud. Algunas edades se encuentran precisadas con números decimales. Se aplicará un redondeo para tener un atributo con solo números enteros

```
trainData["Age"] = floor(trainData["Age"])
testData["Age"] = floor(testData["Age"])
```

Y se observará nuevamente las estadísticas de esta variable en ambos conjuntos de datos para realizar correctamente la discretización.

```
summary(trainData$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   22.00   29.00   29.61   36.50   80.00
```

```
summary(testData$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   23.00   30.00   30.19   35.75   76.00
```

Discretizamos con intervalos.

```
trainData["Rango_Age"] = cut(trainData$Age,
                             breaks = c(-1,10,20,30,40,50,60,70,80),
                             labels = c("0-10","11-20","21-30","31-40",
                                        "41-50","51-60","61-70","71-80"))
testData["Rango_Age"]  <- cut(testData$Age,
                              breaks = c(-1,10,20,30,40,50,60,70,80),
                              labels = c("(0-10","11-20","21-30","31-40",
                                         "41-50","51-60","61-70","71-80"))
```

También se dicotomizarán las variables familiares, dejándolas simplemente en viajaba con/sin hermanos o cónyuges y viajaba con/sin hijos o padres.

```
trainData$SibSp[trainData$SibSp!=0] = 1
trainData$Parch[trainData$Parch!=0] = 1
testData$SibSp[testData$SibSp!=0] = 1
testData$Parch[testData$Parch!=0] = 1

trainData$SibSp = factor(trainData$SibSp,
                         levels=c(0,1),
                         labels= c("Sin hermanos ni cónyuges",
                                   "Con hermanos o cónyuges"))
trainData$Parch = factor(trainData$Parch,
                         levels=c(0,1),
                         labels= c("Sin hijos ni padres",
                                   "Con hijos o padres"))
testData$SibSp = factor(testData$SibSp,
                        levels=c(0,1),
                        labels= c("Sin hermanos ni cónyuges",
                                  "Con hermanos o cónyuges"))
testData$Parch = factor(testData$Parch,
                        levels=c(0,1),
                        labels= c("Sin hijos ni padres",
                                  "Con hijos o padres"))
```

Finalmente, transformaremos algunas variables para que sean de tipo factor.

```
trainData$Sobreviviente = factor(trainData$Survived,
                                 levels = c(0,1),
                                 labels = c("No","Sí"))
trainData$Survived = NULL
trainData$Pclass = factor(trainData$Pclass,
                          levels = c(1,2,3),
                          labels = c("Primera clase",
                                     "Segunda clase",
                                     "Tercera clase"))
trainData$Sex = factor(trainData$Sex,
                       levels = c("male","female"),
                       labels = c("Masculino", "Femenino"))
trainData$Embarked = factor(trainData$Embarked,
                            levels = c("C", "Q", "S"),
                            labels = c("Cherbourg",
                                       "Queenstown",
                                       "Southampton"))

testData$Pclass = factor(testData$Pclass,
                         levels = c(1,2,3),
                         labels = c("Primera clase",
                                    "Segunda clase",
                                    "Tercera clase"))
testData$Sex = factor(testData$Sex,
                      levels = c("male","female"),
                      labels = c("Masculino", "Femenino"))
testData$Embarked = factor(testData$Embarked,
                           levels = c("C", "Q", "S"),
                           labels = c("Cherbourg",
```

```
                                    "Queenstown",
                                    "Southampton"))
```

Nuevo Formato de la data

Nuesta data **test**, ahora presenta 418 registros (filas) que se corresponden a las reseñas y 8 variables (columnas) que los caracterizan.

Atributo	Descripción
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Sex	Sexo
SibSp	Presencia de hermanos/cónyuges a bordo
Parch	Presencia de padres/hijos a bordo
Fare	Tarifa de pasajero
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)
Rango_Age	Rango de Edad

Nuesta data **train**, ahora presenta 891 registros (filas) que se corresponden a las reseñas y 9 variables (columnas) que los caracterizan.

Atributo	Descripción
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Sex	Sexo
Age	Edad
SibSp	Presencia de hermanos/cónyuges a bordo
Parch	Presencia de padres/hijos a bordo
Fare	Tarifa de pasajero
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)
Rango_Age	Rango de Edad
Sobreviviente	Indica si el pasajero sobrevivió o no al accidente SI-NO

Observamos los datos discretizados.

```
head(trainData)
```

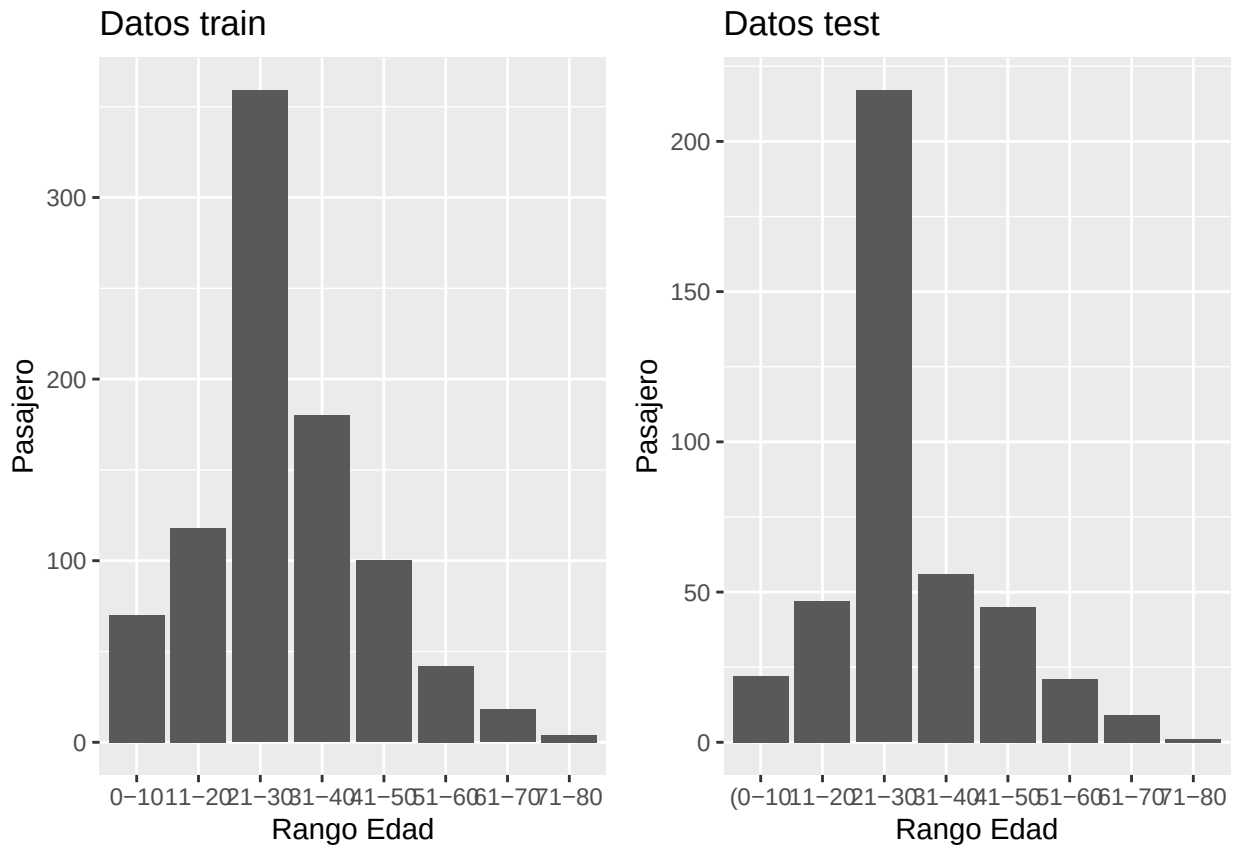
```
##          Pclass       Sex Age                    SibSp               Parch
## 1 Tercera clase Masculino  22  Con hermanos o cónyuges Sin hijos ni padres
## 2 Primera clase  Femenino  38  Con hermanos o cónyuges Sin hijos ni padres
## 3 Tercera clase  Femenino  26 Sin hermanos ni cónyuges Sin hijos ni padres
## 4 Primera clase  Femenino  35  Con hermanos o cónyuges Sin hijos ni padres
## 5 Tercera clase Masculino  35 Sin hermanos ni cónyuges Sin hijos ni padres
## 6 Tercera clase Masculino  29 Sin hermanos ni cónyuges Sin hijos ni padres
##      Fare    Embarked Rango_Age Sobreviviente
## 1  7.2500 Southampton     21-30            No
## 2 71.2833   Cherbourg     31-40            Sí
## 3  7.9250 Southampton     21-30            Sí
## 4 53.1000 Southampton     31-40            Sí
## 5  8.0500 Southampton     31-40            No
## 6  8.4583  Queenstown     21-30            No
```

```
head(testData)
```

```
##          Pclass       Sex Age                    SibSp               Parch
## 1 Tercera clase Masculino  34 Sin hermanos ni cónyuges Sin hijos ni padres
## 2 Tercera clase  Femenino  47  Con hermanos o cónyuges Sin hijos ni padres
## 3 Segunda clase Masculino  62 Sin hermanos ni cónyuges Sin hijos ni padres
## 4 Tercera clase Masculino  27 Sin hermanos ni cónyuges Sin hijos ni padres
## 5 Tercera clase  Femenino  22  Con hermanos o cónyuges  Con hijos o padres
## 6 Tercera clase Masculino  14 Sin hermanos ni cónyuges Sin hijos ni padres
##      Fare    Embarked Rango_Age
## 1  7.8292  Queenstown     31-40
## 2  7.0000 Southampton     41-50
## 3  9.6875  Queenstown     61-70
## 4  8.6625 Southampton     21-30
## 5 12.2875 Southampton     21-30
## 6  9.2250 Southampton     11-20
```

Vemos como los datos se agrupan por segmento de edad.

```
grid.newpage()

plotTrbyAge <- ggplot(trainData,aes(Rango_Age))+geom_bar() +labs(x="Rango Edad", y="Pasajero")+ guides(

plotTebyAge <- ggplot(testData,aes(Rango_Age))+geom_bar() +labs(x="Rango Edad", y="Pasajero")+ guides(f

grid.arrange(plotTrbyAge,plotTebyAge,ncol=2)
```

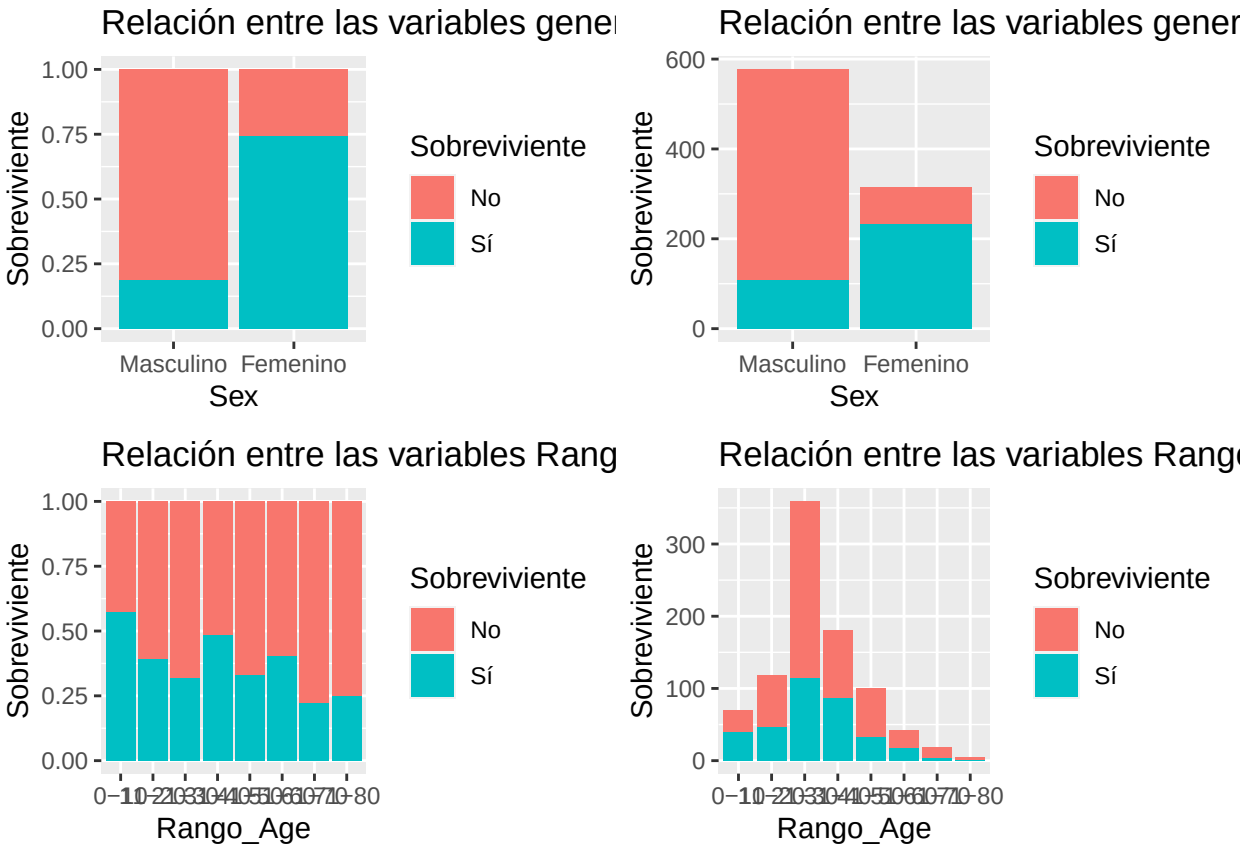


Procesos de análisis visuales del juego de datos

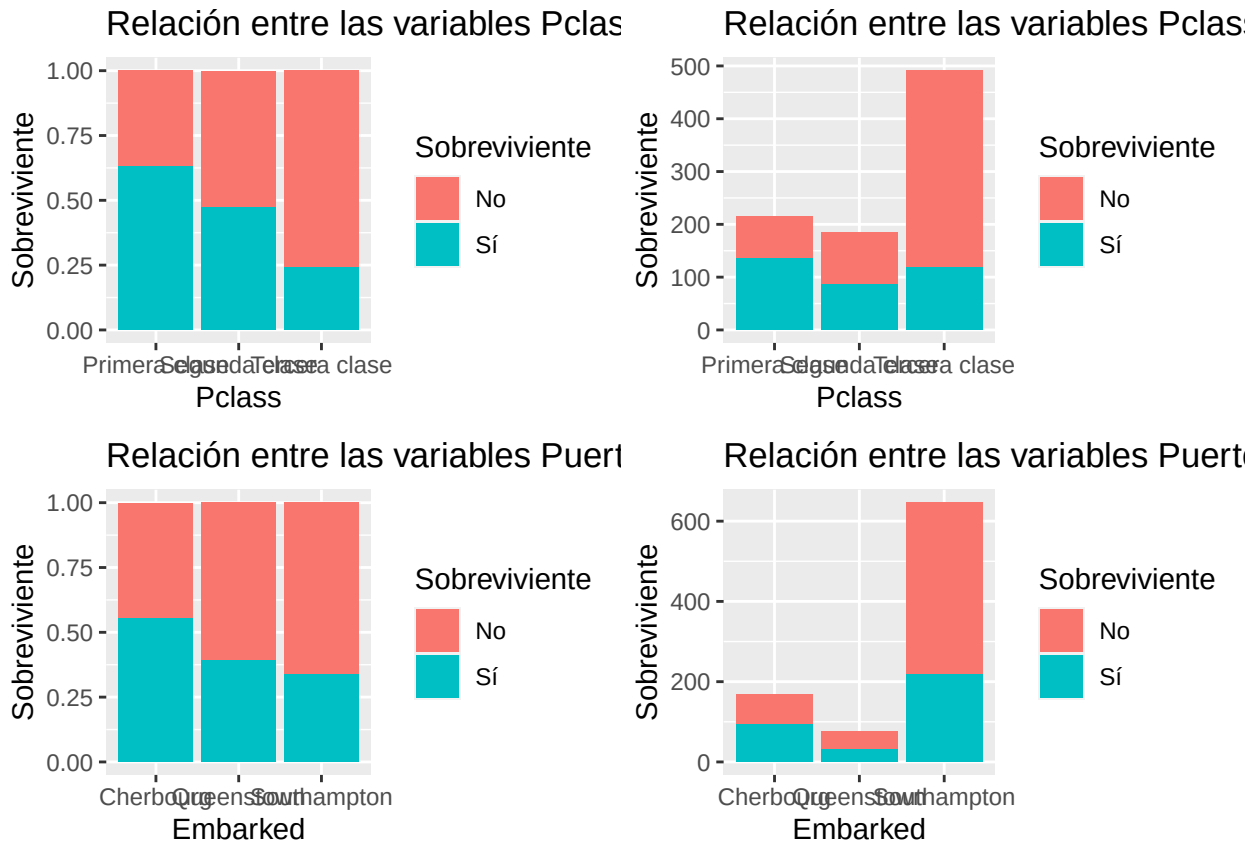
Nos proponemos analizar las relaciones entre las diferentes variables del juego de datos para ver si se relacionan y como.

Visualizamos la relaciones entre las variables categóricas y el atributo sobrevivencia.

```
plotTrbySex1 <- ggplot(trainData,aes(x=Sex,fill=Sobreviviente))+geom_bar(position="fill")+ylab("Sobrevi
plotTrbySex2 <- ggplot(trainData,aes(x=Sex,fill=Sobreviviente))+geom_bar()+ylab("Sobreviviente")+ggtitl
plotTrbyAge1 <- ggplot(trainData,aes(x=Rango_Age,fill=Sobreviviente))+geom_bar(position="fill")+ylab("So
plotTrbyAge2 <- ggplot(trainData,aes(x=Rango_Age,fill=Sobreviviente))+geom_bar()+ylab("Sobreviviente")+g
grid.arrange(plotTrbySex1,plotTrbySex2,plotTrbyAge1,plotTrbyAge2,ncol=2)
```

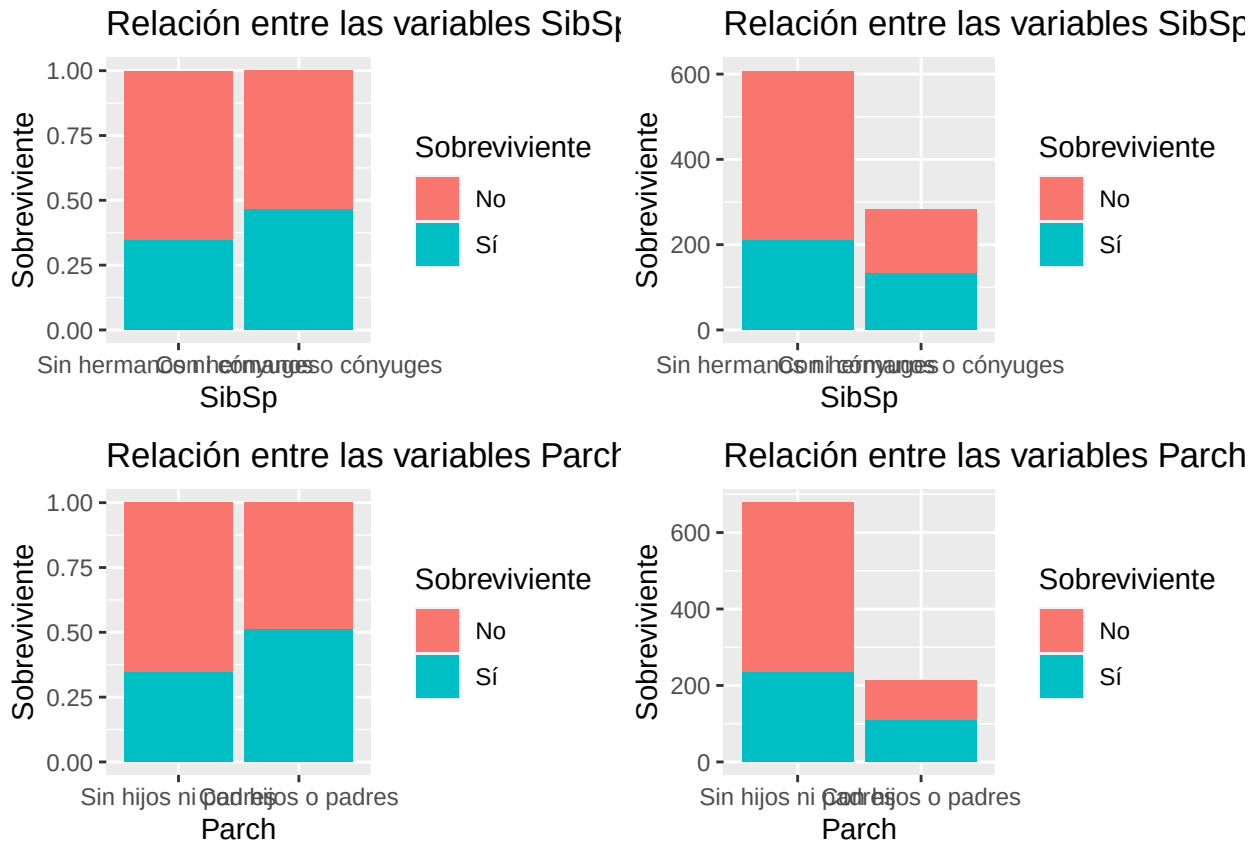



```
plotTrbyCla1 <- ggplot(trainData,aes(x=Pclass,fill=Sobreviviente))+geom_bar(position="fill")+ylab("Sobre
plotTrbyCla2 <- ggplot(trainData,aes(x=Pclass,fill=Sobreviviente))+geom_bar()+ylab("Sobreviviente")+ggt
plotTrbyEmb1 <- ggplot(trainData,aes(x=Embarked,fill=Sobreviviente))+geom_bar(position="fill")+ylab("So
plotTrbyEmb2 <- ggplot(trainData,aes(x=Embarked,fill=Sobreviviente))+geom_bar()+ylab("Sobreviviente")+g
grid.arrange(plotTrbyCla1,plotTrbyCla2,plotTrbyEmb1,plotTrbyEmb2,ncol=2)
```



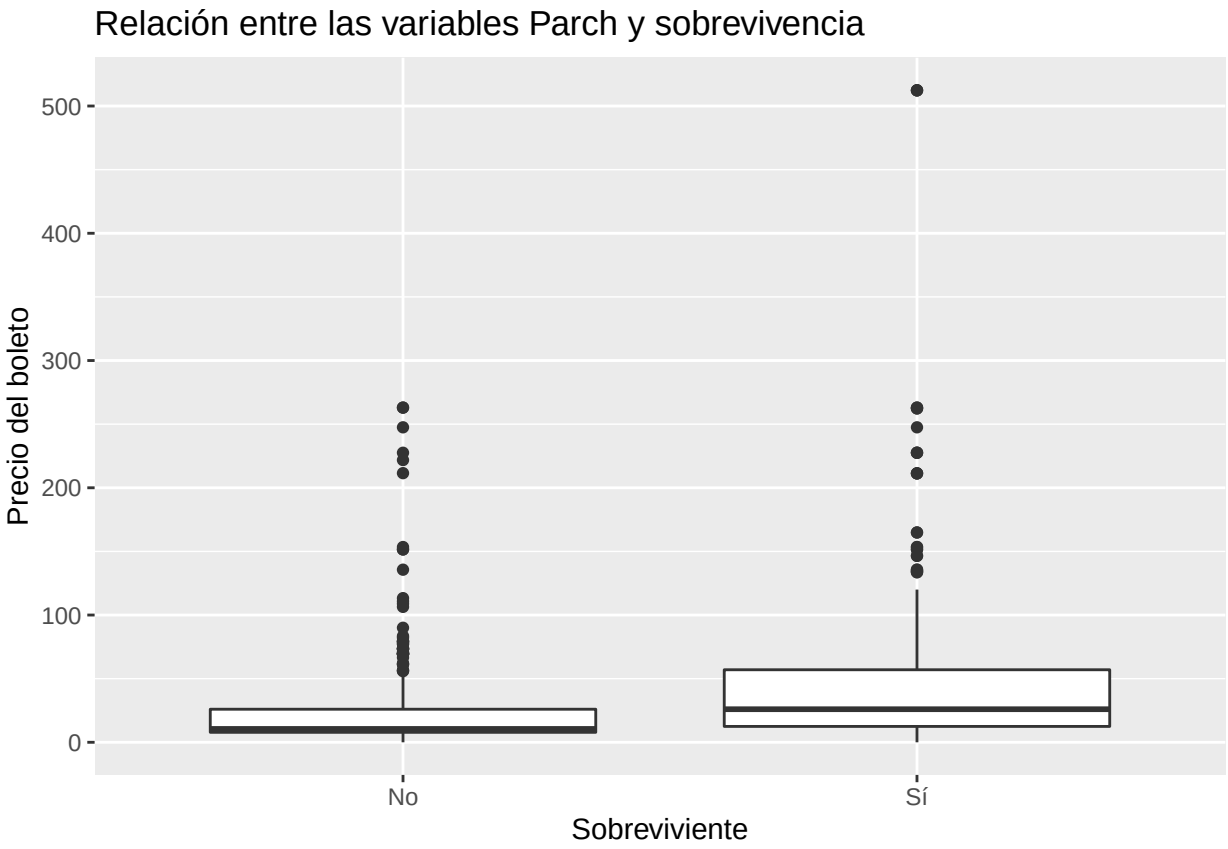
Ahora visualizaremos las variables familiares y su relación con la supervivencia.

```
plotTrbySib1 <- ggplot(trainData,aes(x=SibSp,fill=Sobreviviente))+geom_bar(position="fill")+ylab("Sobrev

plotTrbySib2 <- ggplot(trainData,aes(x=SibSp,fill=Sobreviviente))+geom_bar()+ylab("Sobreviviente")+ggti

plotTrbyPar1 <- ggplot(trainData,aes(x=Parch,fill=Sobreviviente))+geom_bar(position="fill")+ylab("Sobrev

plotTrbyPar2 <- ggplot(trainData,aes(x=Parch,fill=Sobreviviente))+geom_bar()+ylab("Sobreviviente")+ggti

grid.arrange(plotTrbySib1,plotTrbySib2,plotTrbyPar1,plotTrbyPar2,ncol=2)
```



Por último, realizaremos un gráfico de caja para la variable Fare en cada categoría de la variable supervivencia.

```
ggplot(trainData,aes(x=Sobreviviente,y=Fare))+geom_boxplot()+ylab("Precio del boleto")+ggtitle("Relació
```



Conclusiones previas.

De acuerdo a los graficos obtenidos, Existe una grán posibilidad de sobrevivir, si el pasajero, es mujer, es menor de 60 años, tiene un ticket de primera clase, embarca en el puerto de Cherbourg, tiene familiares a bordo y/o si la tarifa pagada fue de mayor precio.

Fase de Modelado.

En esta fase se construirán y evaluarán varios modelos, se probarán algoritmos y técnicas hasta encontrar un modelo adecuado.

Las conclusiones preliminares de la sección anterior fueron realiadas en función de la observación de los gráficos generados. No obstante, es necesario aplicar procedimientos de inferencia para determinar si las diferencias visualizadas son estadísticamente significativas. Nos concentraremos en 3 procedimientos:

1) Se realizarán pruebas para comparar las medias de la tarifa del pasajero en cada categoría de la variable supervivencia y las medias de la edad del pasajero en cada categoría de la variable supervivencia.

2) Se realizará una prueba chicuadrado para determinar dependencia entre el sexo y la sobrevivencia, entre el lugar de embarque y la sobrevivencia, y entre las variables familiares y la sobrevivencia.

3) Se entrenará un modelo de árbol de decisión para predecir la sobrevivencia de los pasajeros en función de las variables analizadas.

Respecto de la prueba de comparación de medias, es necesario en primer lugar observar si los datos son normales y homocedásticos para determinar si puede aplicarse la prueba t o, en caso de que no se satisfagan las condiciones, aplicar algún procedimiento no paramétrico.

Se evaluará la normalidad de la variable Fair en cada categoría mediante la prueba de Shapiro Wilk.

```
shapiro.test(trainData$Fare[trainData$Sobreviviente=="Sí"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  trainData$Fare[trainData$Sobreviviente == "Sí"]
## W = 0.59673, p-value < 2.2e-16
```

```
shapiro.test(trainData$Fare[trainData$Sobreviviente=="No"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  trainData$Fare[trainData$Sobreviviente == "No"]
## W = 0.51304, p-value < 2.2e-16
```

Puede observarse que el valor p en ambos test arroja un valor muy cercano a cero. En este caso la hipótesis de normalidad es rechazada en ambos casos.

Procederemos a evaluar la homocedasticidad. Como los datos no son normales, se ha elegido en este caso la prueba de Leneve.

```
leveneTest(trainData$Fare, trainData$Sobreviviente)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   1    45.1 3.337e-11 ***
##       889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El valor p de la prueba de Leneve indica que la hipótesis nula de homocedasticidad es rechazada.

Como los datos son heterocedásticos y no normales, se realizará una prueba U de Wilcoxon - Mann - Whitney para determinar diferencia significativa entre el precio de la tarifa pagada por los sobrevivientes y quienes murieron en la tragedia.

```
wilcox.test(trainData$Fare[trainData$Sobreviviente=="Sí"],
            trainData$Fare[trainData$Sobreviviente=="No"])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  trainData$Fare[trainData$Sobreviviente == "Sí"] and trainData$Fare[trainData$Sobreviviente ==
## W = 129952, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

El valor p de la prueba U indica que la hipótesis de igual distribución entre ambos grupos es rechazada.

Por lo tanto, existe evidencia suficiente para descartar la hipótesis que sobrevivientes y fallecidos hayan pagado, en promedio, la misma tarifa.

Realizaremos el mismo proceso para la variable Age.

```
shapiro.test(trainData$Age[trainData$Sobreviviente=="Sí"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  trainData$Age[trainData$Sobreviviente == "Sí"]
## W = 0.98376, p-value = 0.0006697
```

```
shapiro.test(trainData$Age[trainData$Sobreviviente=="No"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  trainData$Age[trainData$Sobreviviente == "No"]
## W = 0.95871, p-value = 2.783e-11
```

La edad tampoco sigue una distribución normal. Se prueba la homocedasticidad:

```
leveneTest(trainData$Age, trainData$Sobreviviente)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   1  7.0445 0.008093 **
##       889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A un nivel de significancia de 0,05, los datos son heterocedásticos. Se realiza la prueba U:

```
wilcox.test(trainData$Age[trainData$Sobreviviente=="Sí"],
            trainData$Age[trainData$Sobreviviente=="No"])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  trainData$Age[trainData$Sobreviviente == "Sí"] and trainData$Age[trainData$Sobreviviente == "
## W = 86302, p-value = 0.04235
## alternative hypothesis: true location shift is not equal to 0
```

A un nivel de significancia de 0,05, existen diferencias en la distribución de la edad entre sobrevivientes y fallecidos.

Procederemos ahora a evaluar con una prueba chicuadardo la relación entre el sexo y la supervivencia

```
chisq.test(trainData$Sobreviviente,trainData$Sex)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  trainData$Sobreviviente and trainData$Sex
## X-squared = 260.72, df = 1, p-value < 2.2e-16
```

El valor p cercano a cero indica que se rechaza la hipótesis nula de independencia entre ambas variables. Es decir, existe relación estadísticamente significativa entre el sexo de la persona y su probabilidad de supervivencia.

Cabe destacar que en este caso es posible realizar la prueba chicuadrado, ya que los valores esperados en cada celda de la tabla de contingencia es mayor a 5:

```
chisq.test(trainData$Sobreviviente,trainData$Sex)$expected
```

```
##                       trainData$Sex
## trainData$Sobreviviente Masculino Femenino
##                      No  355.5253 193.4747
##                      Sí  221.4747 120.5253
```

Por lo que no fue necesario reemplazar esta prueba por el test exacto de Fisher.

Se realiza la prueba chicuadrado para las demás variabes categóricas (para no extender este documento, se han omitido las comprobaciones sobre los valores esperados)

```
chisq.test(trainData$Sobreviviente,trainData$SibSp)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  trainData$Sobreviviente and trainData$SibSp
## X-squared = 11.456, df = 1, p-value = 0.0007128
```

```
chisq.test(trainData$Sobreviviente,trainData$Parch)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  trainData$Sobreviviente and trainData$Parch
## X-squared = 18.656, df = 1, p-value = 1.565e-05
```

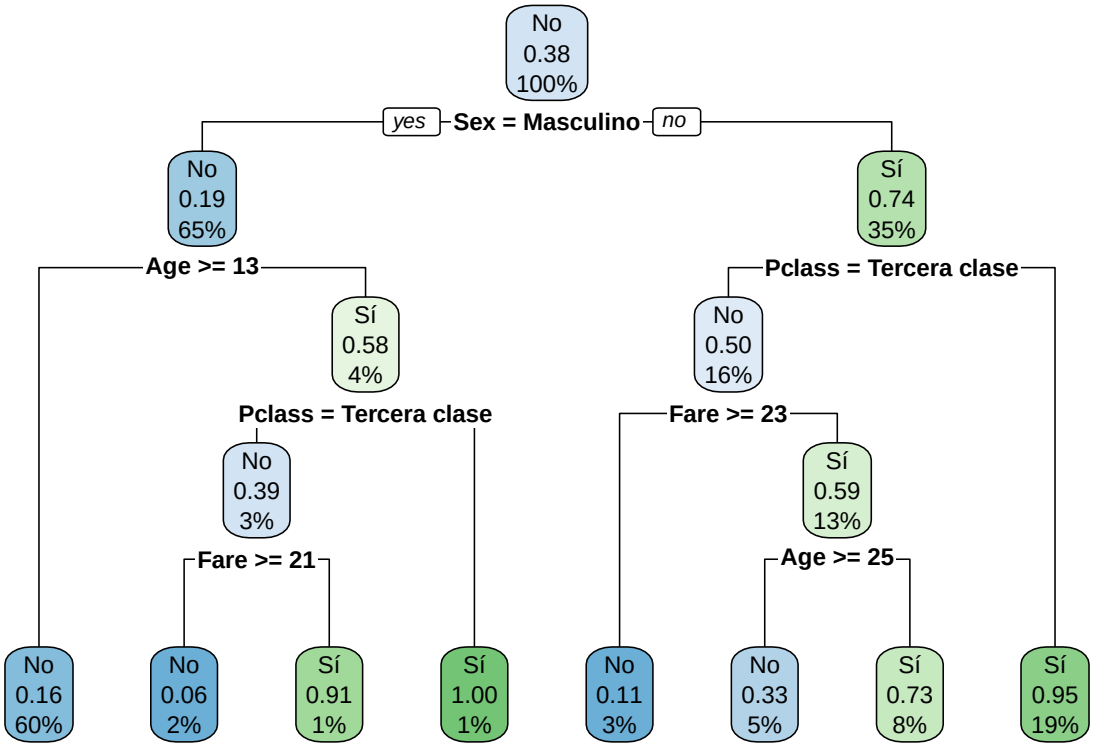
```
chisq.test(trainData$Sobreviviente,trainData$Embarked)
```

```
##
##  Pearson's Chi-squared test
##
## data:  trainData$Sobreviviente and trainData$Embarked
## X-squared = 25.964, df = 2, p-value = 2.301e-06
```

Puede observarse que se rechaza la hipótesis nula en las 3 pruebas, indicando que hay relación entre la supervivencia y las 3 variables analizadas.

Por último, entrenaremos un modelo de árbol de decisión. Cabe señalar que dadas las características categóricas de la mayoría de las variables, se optó por este procedimiento en vez de realizar una regresión logística.

```
trainData2 = select(trainData,-Rango_Age)
arbol = rpart(Sobreviviente ~ ., trainData2)
rpart.plot(arbol)
```



El modelo entrenado contiene las variables sexo, edad, clase y tarifa. El árbol resultante plantea las siguientes reglas de decisión:

Sexo femenino y no es de tercera clase –> Sobrevive Sexo femenino de tercera clase que pagó menos de 23 de tarifa y tiene menos de 25 años –> Sobrevive Sexo masculino menor de 13 años que no es de tercera clase –> Sobrevive Sexo masculino de tercera clase que pagó menos de 21 de tarifa –> Sobrevive

La natriz de confusión del modelo sobre los datos de entrenamiento indica que el modelo logra clasificar correctamete el 84,96% de los casos, con una sensibilidad del 94,54% y 69,59%.

```
predic = predict(arbol,trainData,type="class")
confusionMatrix(predic, trainData$Sobreviviente)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No  Sí
```

```
##         No 519 104
##         Sí  30 238
##
##                Accuracy : 0.8496
##                  95% CI : (0.8244, 0.8725)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6685
##
##  Mcnemar's Test P-Value : 2.859e-10
##
##             Sensitivity : 0.9454
##             Specificity : 0.6959
##          Pos Pred Value : 0.8331
##          Neg Pred Value : 0.8881
##              Prevalence : 0.6162
##          Detection Rate : 0.5825
##    Detection Prevalence : 0.6992
##       Balanced Accuracy : 0.8206
##
##        'Positive' Class : No
##
```

Claramente es un modelo que puede perfeccionarse, ya sea aplicando otra técnica para crear árboles de decisión o buscando otro modelo má complejo, como redes neuronales, por ejemplo. Sin embargo, consideramos que este modelo, además de todo lo demás realizado, cumple con el objetivo de la práctica, por lo que lo aplicaremos al conjunto de prueba.

Resolución del problema

Se aplicará el modelo de árbol de decisión generado al conjunto de prueba. Para ello, es necesario

Como este conjunto de datos no tiene el valor real de la sobrevivencia, no podremos evaluarlo mediante una matriz de confusión, pero sí se generarán tablas de contingencia para observar las predicciones realizadas.

```
testData$predic = predict(arbol, testData, type="class")
table(testData$predic)
```

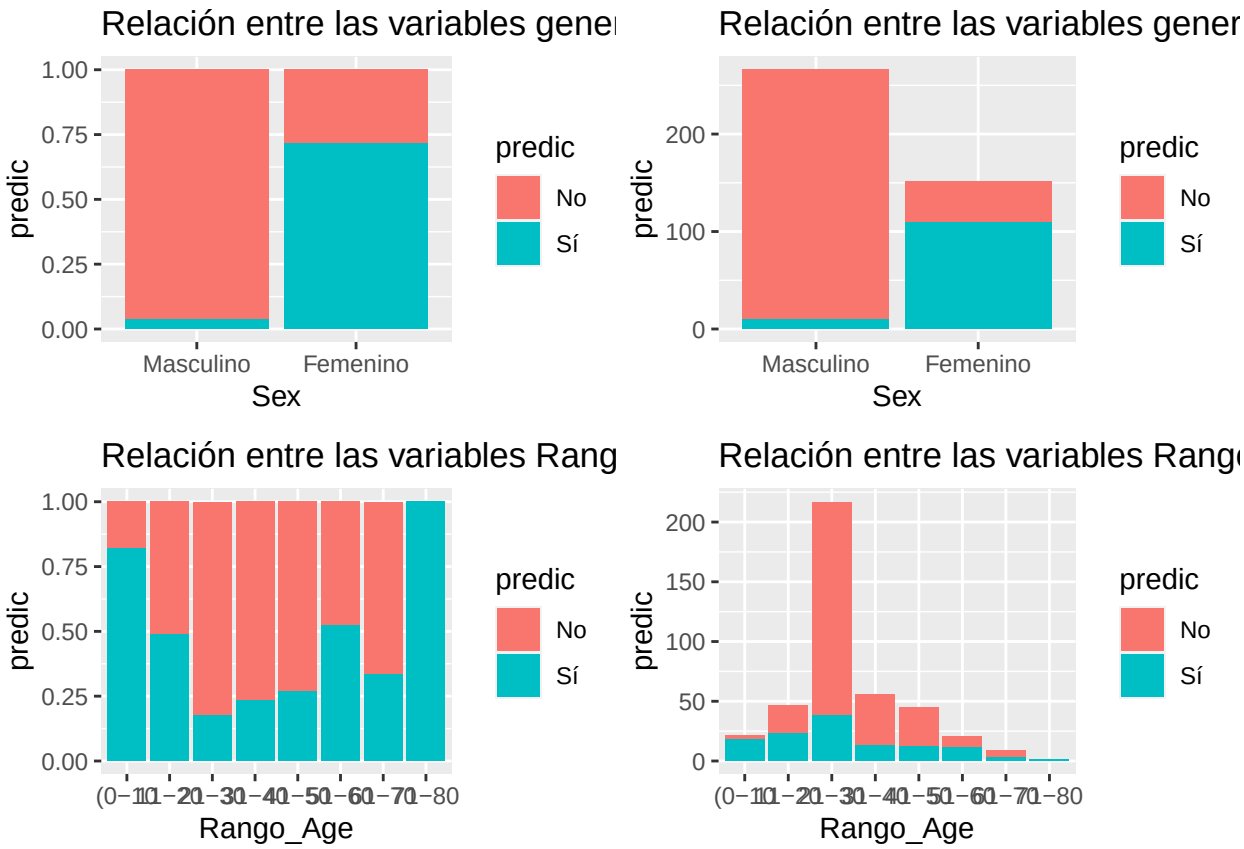
```
##
##  No  Sí
## 299 119
```

Según la predicción, 119 personas del conjunto de prueba sobrevivirían y 299 no.

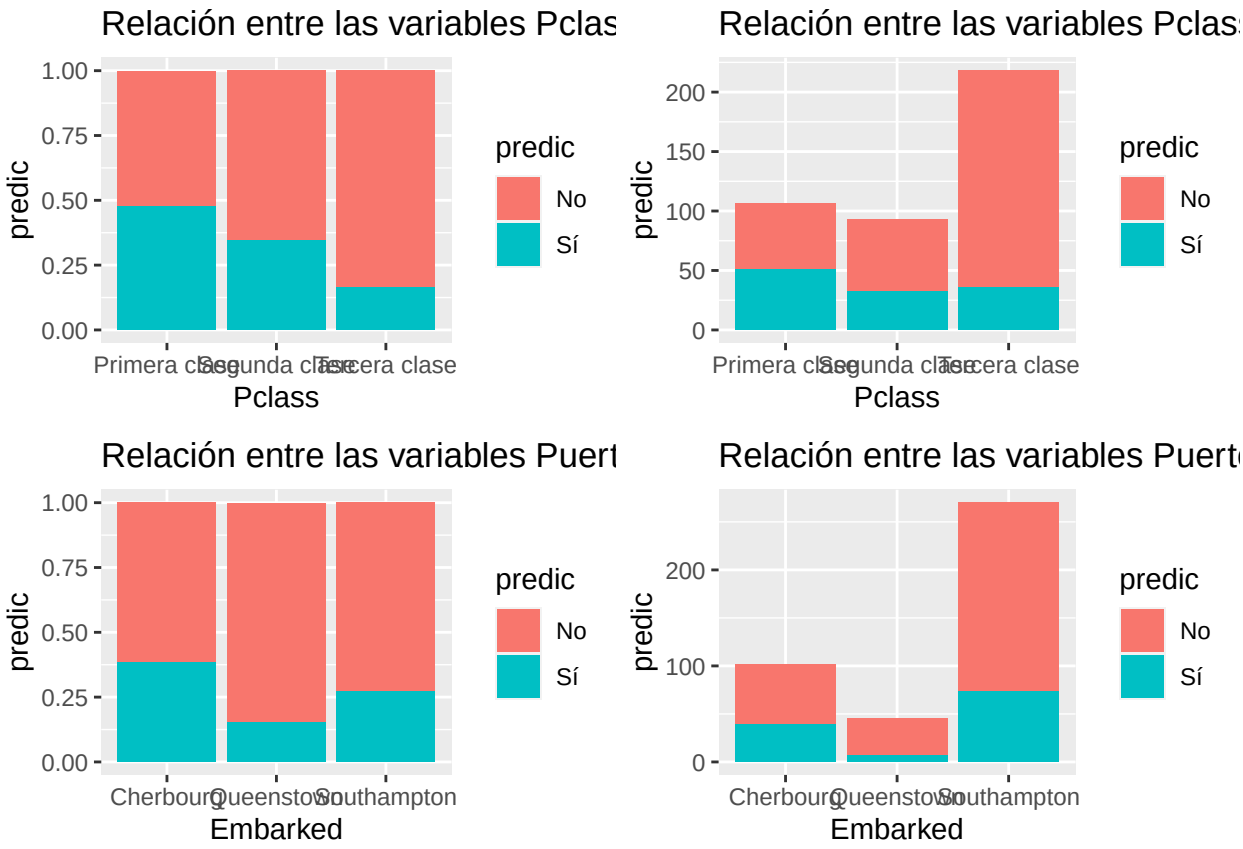
Los siguientes gráficos muestran la distribución de la predicción de sobrevivientes y fallecidos en las diferentes variables categóricas

```
plotTrbySex1 <- ggplot(testData,aes(x=Sex,fill=predic))+geom_bar(position="fill")+ylab("predic")+ggtitl
```

```
plotTrbySex2 <- ggplot(testData,aes(x=Sex,fill=predic))+geom_bar()+ylab("predic")+ggtitle("Relación enti
plotTrbyAge1 <- ggplot(testData,aes(x=Rango_Age,fill=predic))+geom_bar(position="fill")+ylab("predic")+g
plotTrbyAge2 <- ggplot(testData,aes(x=Rango_Age,fill=predic))+geom_bar()+ylab("predic")+ggtitle("Relaci
grid.arrange(plotTrbySex1,plotTrbySex2,plotTrbyAge1,plotTrbyAge2,ncol=2)
```




```
plotTrbyCla1 <- ggplot(testData,aes(x=Pclass,fill=predic))+geom_bar(position="fill")+ylab("predic")+ggt
plotTrbyCla2 <- ggplot(testData,aes(x=Pclass,fill=predic))+geom_bar()+ylab("predic")+ggtitle("Relación
plotTrbyEmb1 <- ggplot(testData,aes(x=Embarked,fill=predic))+geom_bar(position="fill")+ylab("predic")+g
plotTrbyEmb2 <- ggplot(testData,aes(x=Embarked,fill=predic))+geom_bar()+ylab("predic")+ggtitle("Relació
grid.arrange(plotTrbyCla1,plotTrbyCla2,plotTrbyEmb1,plotTrbyEmb2,ncol=2)
```




```
plotTrbySib1 <- ggplot(testData,aes(x=SibSp,fill=predic))+geom_bar(position="fill")+ylab("predic")+ggti

plotTrbySib2 <- ggplot(testData,aes(x=SibSp,fill=predic))+geom_bar()+ylab("predic")+ggtitle("Relación e

plotTrbyPar1 <- ggplot(testData,aes(x=Parch,fill=predic))+geom_bar(position="fill")+ylab("predic")+ggti

plotTrbyPar2 <- ggplot(testData,aes(x=Parch,fill=predic))+geom_bar()+ylab("predic")+ggtitle("Relación e

grid.arrange(plotTrbySib1,plotTrbySib2,plotTrbyPar1,plotTrbyPar2,ncol=2)
```

Puede visualizarse que el modelo predice que los sobrevivientes serán principalmente mujeres, menores de 20 años, que poseen boletos de primera clase, con familiares a bordo y/o que o embarcaron en Queenstown.

Finalmente, exportaremos ambos dataframes en archivos csv

```
write.csv(trainData,"trainMod.csv")
write.csv(testData,"testMod.csv")
```

Tabla de contribuciones

Tabla con los datos de entrenamiento: trainMod.csv

Tabla con los datos de pruebas: testMod.csv
