

Tipología y ciclo de vida de los datos: Práctica 2

Azucena González (azucenagm) y Jesús Márquez (jmarquez01)

Mayo 2019

Índice

Introducción	2
Descripción del dataset	2
Carga de los datos	3
Integración y selección de los datos	5
Integración	5
Selección de los datos de interés	7
Reducción de la dimensionalidad	7
Reducción de cantidad	8
Limpieza de los datos	8
Tratamiento de elementos vacíos y ceros	9
Revisión de los datos	9
Tratamiento	9
Tratamiento de valores extremos	12
Discretización	17
Análisis de los datos	18
Análisis descriptivo visual	18
Variable Survived	18
Variable Pclass	19
Variable Sex	20
Variable Age	21
Variable SibSp	22
Variable Parch	23
Variable Fare	24
Variable Embarked	25
Correlación entre variables cuantitativas	26
Relación de Survived con otras variables	28
Relación entre variables categóricas	31
Planificación	35
Comprobación de normalidad y homocedasticidad	36
Aplicación de pruebas estadísticas	36
Árbol de decisión	36
Modelo de regresión logística	37
Random forest	39
Representación de los resultados	42
Conclusiones	45
Código fuente	46
Bibliografía	46
Contribuciones	47

Introducción

El hundimiento del RMS Titanic es uno de los naufragios más infames de la historia. El 15 de abril de 1912, durante su viaje inaugural, el Titanic se hundió después de chocar con un iceberg, matando a 1502 de 2224 pasajeros y tripulantes. Esta tragedia conmocionó a la comunidad internacional y condujo a mejores regulaciones de seguridad para los buques, dado que una de las principales razones de tal pérdida de vidas fue que no había suficientes botes salvavidas para los pasajeros y la tripulación.

El objetivo del siguiente **estudio KDD** es realizar un análisis que permita **determinar qué tipos de personas podrían sobrevivir** al hundimiento. Se aplicarán, por tanto, varias técnicas de minería de datos y *machine learning* para obtener un modelo que permita predecir con un nivel de calidad aceptable qué pasajeros sobrevivirían a la tragedia.

Descripción del dataset

Se ha obtenido el conjunto de datos disponible en Kaggle (<https://www.kaggle.com>) con el listado de pasajeros del Titanic y si sobrevivieron o no al naufragio. Los datos recogidos contienen los siguientes campos:

- PassengerId: identificador numérico (entero) para cada pasajero.
- Survived: que toma dos posibles valores e indica si el pasajero sobrevivió. 0 significa que no sobrevivió, 1 que sí lo hizo.
- Pclass: número entero que indica la clase en la que viaja el pasajero. Se ha codificado como 1 para primera clase, 2 para segunda clase y 3 para tercera clase.
- Name: nombre del pasajero.
- Sex: sexo del pasajero. Se indica male para hombres y female para mujeres.
- Age: número entero que representa la edad del pasajero.
- SibSp: número de hermanos/cónyuges que se encontraban a bordo.
- Parch: número de padres/hijos que se encontraban a bordo.
- Ticket: identificador del billete.
- Fare: numérico con decimales que representa la tarifa del billete.
- Cabin: número de cabina.
- Embarked: puerto de embarque, habiéndose codificado como C para Cherbourg, Q para Queenstown y S para Southampton.

El dataset se encuentra dividido en dos ficheros:

- **titanic.train.csv** o conjunto de entrenamiento, que contiene 892 registros y que suele emplearse para entrenar las técnicas supervisadas de data mining que se utilicen durante el estudio.
- **titanic.test.csv** o conjunto de prueba, con 419 observaciones, que suele utilizarse en algunos algoritmos supervisados para evaluar la calidad del modelo construido. Este fichero no contiene el campo *Survived*, pero puede encontrarse esta información en **titanic.test.solution.csv**.

Antes de iniciar la carga de los ficheros correspondientes, se ejecutan las librerías que se van a usar durante el análisis.

```
# Se comprueba si los paquetes de las librerías a usar están descargados. Si no, se
# procede a su descarga e instalación

# Librerías con utilidades para matrices de correlación
if(!"car" %in% installed.packages()) install.packages("car", depend=TRUE)
library(car)

if(!"corrplot" %in% installed.packages()) install.packages("corrplot", depend=TRUE)
library(corrplot)

# Visualización de gráficas
if(!"ggplot2" %in% installed.packages()) install.packages("ggplot2", depend=TRUE)
```

```

library(ggplot2)

# Paleta de colores para gráficas
if(!"RColorBrewer" %in% installed.packages()) install.packages("RColorBrewer",
                                                                depend=TRUE)
library(RColorBrewer)

# Funciones y utilidades para el tratamiento de data frames
if(!"dplyr" %in% installed.packages()) install.packages("dplyr", depend=TRUE)
library(dplyr)

# Funciones relacionadas con curvas ROC
if(!"pROC" %in% installed.packages()) install.packages("pROC", depend=TRUE)
library(pROC)

# Utilidades para formateo avanzado de tablas
if(!"kableExtra" %in% installed.packages()) install.packages("kableExtra", depend=TRUE)
library(kableExtra)

# Funciones relacionadas con el algoritmo kNN (VIM)
if(!"VIM" %in% installed.packages()) install.packages("VIM", depend=TRUE)
library(VIM)

# Funciones relacionadas con paquete caret
if(!"caret" %in% installed.packages()) install.packages("caret", depend=TRUE)
library(caret)

# Funciones relacionadas con los árboles de decisión
if(!"C50" %in% installed.packages()) install.packages("C50", depend=TRUE)
library(C50)

# Funciones para la ejecución del algoritmo random forest
if(!"randomForest" %in% installed.packages()) install.packages("randomForest",
                                                                depend=TRUE)
library(randomForest)

```

Carga de los datos

Se realiza la carga de los dos conjuntos de datos, indicando que la primera línea es la cabecera de datos y que deben eliminarse los espacios en blanco que pueda haber entre campos. También se realiza la carga del fichero que contiene los valores de la variable Survived para el conjunto de datos de prueba.

```

# Datos para entrenamiento
trainData <- read.csv("titanic.train.csv", header=TRUE, strip.white=TRUE,
                     stringsAsFactors=FALSE)

# Datos para test
testData <- read.csv("titanic.test.csv", header=TRUE, strip.white=TRUE,
                    stringsAsFactors=FALSE)

# Datos con el valor de la variable Survived para los datos de test
testSolution <- read.csv("titanic.test.solution.csv", header=TRUE, strip.white=TRUE,
                        stringsAsFactors=FALSE)

```

```
# Se visualizan los primeros registros para comprobar que la carga ha sido correcta
kable(head(trainData)) %>% kable_styling(latex_options=c("striped", "scale_down"))
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500		S
6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583		Q

```
kable(head(testData)) %>% kable_styling(latex_options=c("striped", "scale_down"))
```

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292		Q
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000		S
894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875		Q
895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625		S
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875		S
897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250		S

```
kable(head(testSolution)) %>% kable_styling(latex_options=c("striped"))
```

PassengerId	Survived
892	0
893	1
894	0
895	0
896	1
897	0

```
# Se muestra la estructura de ambos datasets para revisar los tipos de datos asignados
# por defecto
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" ...
```

```
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 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" ...
```

```
str(testSolution)
```

```
## 'data.frame':  418 obs. of  2 variables:
## $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
## $ Survived   : int  0 1 0 0 1 0 1 0 1 0 ...
```

Integración y selección de los datos

Integración

Aunque inicialmente se nos facilita el conjunto total de datos separado entre el conjunto de pruebas y el de entrenamiento, vamos a proceder a tratar ambos conjuntamente para su preprocesado y limpieza. Posteriormente, para realizar la evaluación de varios modelos, se separarán en varios grupos (*k-fold cross validation*).

En primer lugar, debe incorporarse al conjunto de datos de prueba la variable objeto del análisis, Survived, que se encuentra en el dataset que hemos denominado *testSolution*. Después se unificarán en un único dataframe los dos conjuntos de datos.

```
# Comprobamos que el PassengerId no se repite ni dentro de cada grupo ni en ambos
# conjuntamente: es un identificador que debería ser único
```

```
# Suma del número de registros de ambos grupos
length(trainData$PassengerId) + length(testData$PassengerId)
```

```
## [1] 1309
```

```
# Total de registros únicos
length(unique(append(trainData$PassengerId, testData$PassengerId)))
```

```
## [1] 1309
```

El número de PassengerId únicos coincide con el total de registros de ambos *dataframes*, por lo que no hay observaciones repetidas.

```
# Se realiza un inner join por PassengerId
testDataSolution <- merge(x=testData, y=testSolution, by="PassengerId")

# Se visualizan los primeros registros para comprobar que el join ha sido correcto
kable(head(testDataSolution)) %>% kable_styling(latex_options=c("striped", "scale_down"))
```

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292		Q	0
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000		S	1
894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875		Q	0
895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625		S	0
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875		S	1
897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250		S	0

```
# Se muestra la estructura del dataset para comprobar que es correcta (mismas variables y del
# mismo tipo)
str(testDataSolution)
```

```
## 'data.frame':   418 obs. of  12 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" ...
## $ Survived   : int    0 1 0 0 1 0 1 0 1 0 ...
```

```
# Por último, se unen los dos conjuntos de datos en un único data frame
titanicData <- rbind(trainData, testDataSolution)
```

```
# A modo de comprobación, se visualizan los primeros registros
kable(head(titanicData)) %>% kable_styling(latex_options=c("striped", "scale_down"))
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500		S
6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583		Q

```
# La estructura del data frame resultante
str(titanicData)
```

```
## 'data.frame':   1309 obs. of  12 variables:
## $ PassengerId: int   1 2 3 4 5 6 7 8 9 10 ...
## $ Survived   : int    0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass     : int    3 1 3 1 3 3 1 3 3 2 ...
## $ Name       : chr    "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex        : chr    "male" "female" "female" "female" ...
## $ Age        : num    22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp      : int    1 1 0 1 0 0 0 3 0 1 ...
## $ Parch      : int    0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket     : chr    "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare       : num    7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin      : chr    "" "C85" "" "C123" ...
## $ Embarked   : chr    "S" "C" "S" "S" ...
```

```
# Y que no hay passengerId repetidos
sum(duplicated(titanicData$passengerId))
```

```
## [1] 0
```

Se revisan los tipos asignados a cada variable: Survived, Pclass, Sex y Embarked son variables categóricas, por lo que se deben declarar como factores.

```
# Se convierten a factores las variables indicadas
titanicData$Survived <- factor(titanicData$Survived, levels=c(0, 1),
                               labels=c("No", "Si"))
titanicData$Pclass <- as.factor(titanicData$Pclass)
titanicData$Sex <- as.factor(titanicData$Sex)
titanicData$Embarked <- as.factor(titanicData$Embarked)
```

```
# Se comprueba el cambio de tipos
```

```
str(titanicData)
```

```
## 'data.frame': 1309 obs. of 12 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Survived : Factor w/ 2 levels "No","Si": 1 2 2 2 1 1 1 1 2 2 ...
## $ Pclass : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr "" "C85" "" "C123" ...
## $ Embarked : Factor w/ 4 levels "", "C", "Q", "S": 4 2 4 4 4 3 4 4 4 2 ...
```

Selección de los datos de interés

Reducción de la dimensionalidad

Tras la revisión inicial del *dataset* y las variables que lo forman, se decide eliminar la relativa al nombre del pasajero, ya que no debería tener relevancia alguna en el estudio. Eliminamos también `PassengerId`, pues no se trata de ninguna característica de los pasajeros, sino de un identificador asignado a posteriori, con el único objetivo de tener identificada cada una de las observaciones del dataset.

```
# Se eliminan las variables Name y PassengerID
```

```
titanicData <- select(titanicData, -c("PassengerId", "Name"))
str(titanicData)
```

```
## 'data.frame': 1309 obs. of 10 variables:
## $ Survived: Factor w/ 2 levels "No","Si": 1 2 2 2 1 1 1 1 2 2 ...
## $ Pclass : Factor w/ 3 levels "1","2","3": 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 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr "" "C85" "" "C123" ...
## $ Embarked: Factor w/ 4 levels "", "C", "Q", "S": 4 2 4 4 4 3 4 4 4 2 ...
```

Con el objetivo de ver si es posible descartar alguna de las variables numéricas, procedemos a aplicar un análisis de componentes principales. Mediante esta técnica se puede detectar qué variables aportan un porcentaje de varianza mayor y son, por tanto, más significativas.

```
# Se ejecuta PCA
```

```
titanic.pca <- prcomp(na.omit(select(titanicData, c("Age", "SibSp", "Parch", "Fare"))),
                     center=TRUE, scale=TRUE)
summary(titanic.pca)
```

```
## Importance of components:
##              PC1      PC2      PC3      PC4
## Standard deviation 1.2510 1.0912 0.7968 0.7807
## Proportion of Variance 0.3912 0.2977 0.1587 0.1524
## Cumulative Proportion 0.3912 0.6889 0.8476 1.0000
```

Aunque los dos primeros componentes contribuyen a una mayor proporción de varianza (casi un 70 %), los dos restantes aportan un porcentaje lo suficientemente elevado (algo más del 30 %) y parejo entre sí, como para tomar la decisión de mantener todas las variables en la construcción del modelo.

No obstante, es probable que tras un estudio más detallado de los datos en los siguientes apartados, se decida eliminar algún atributo más. Además, se podrán descartar puntualmente variables en la aplicación de alguna de las técnicas estadísticas del análisis, ya sea para mejorar su desempeño o por incompatibilidad del tipo de dato con el algoritmo en uso.

Reducción de cantidad

Dado que el número de registros del data frame no es muy elevado, no se plantea una reducción de la cantidad de registros.

Limpieza de los datos

Como paso previo a la aplicación de cualquier técnica de análisis de datos, es preciso revisar el *dataset* suministrado con el objetivo de encontrar errores e incongruencias y homogeneizar el contenido de las variables que lo conforman.

Mediante la función `summary` visualizamos un resumen completo de los datos:

- Para las variables cuantitativas, se muestran los valores mínimos/máximos, la media, la mediana y los cuartiles primero y tercero.
- Para las variables cualitativas definidas como factores, la frecuencia de valores por categoría.

```
# Resumen de las variables del dataset
summary(titanicData)
```

```
## Survived Pclass      Sex      Age      SibSp
## No:815   1:323   female:466   Min.   : 0.17   Min.   :0.0000
## Si:494   2:277   male :843   1st Qu.:21.00   1st Qu.:0.0000
##          3:709           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      Ticket      Fare      Cabin
## Min.   :0.000   Length:1309   Min.   : 0.000   Length:1309
## 1st Qu.:0.000   Class :character   1st Qu.: 7.896   Class :character
## Median :0.000   Mode  :character   Median :14.454   Mode  :character
## Mean    :0.385           Mean    :33.295
## 3rd Qu.:0.000           3rd Qu.:31.275
## Max.    :9.000           Max.    :512.329
##          NA's      :1
## Embarked
##      : 2
## C:270
## Q:123
## S:914
##
##
```

Con esta breve revisión ya pueden apreciarse varios aspectos que deben tratarse antes de iniciar cualquier proceso analítico como, por ejemplo, la existencia de valores perdidos (NA's) en algunas de las variables

(Age y Fare) y de valores vacíos (Embarked). También llaman la atención algunos valores encontrados en las variables numéricas, como los valores mínimo (0) y máximo (512.3292) de Fare, éste último muy alejado de la mediana.

Tratamiento de elementos vacíos y ceros

Revisión de los datos

Comprobamos el total de datos que presentan valores vacíos (NA y no informados) y a cero por variable:

```
# Número de observaciones con valores NA
cat("\nNúmero de observaciones con valores NA\n")
```

```
##
## Número de observaciones con valores NA
```

```
colSums(is.na(titanicData))
```

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

```
# Número de observaciones con valores vacíos
cat("\nNúmero de observaciones con valores vacíos\n")
```

```
##
## Número de observaciones con valores vacíos
```

```
colSums(titanicData=="", na.rm=TRUE)
```

```
## Survived   Pclass      Sex      Age   SibSp   Parch   Ticket   Fare
##          0         0         0         0       0       0         0         0
##   Cabin Embarked
##    1014         2
```

```
# Número de registros con ceros
cat("\nNúmero de registros con ceros\n")
```

```
##
## Número de registros con ceros
```

```
colSums(titanicData==0, na.rm=TRUE)
```

```
## Survived   Pclass      Sex      Age   SibSp   Parch   Ticket   Fare
##          0         0         0         0     891    1002         0      17
##   Cabin Embarked
##          0         0
```

Hay valores a cero en las variables numéricas SibSp, Parch y Fare. En las dos primeras (número de hermanos/cónyuges y número de padres/hijos) este valor es razonable y se asumirán estos valores como correctos. En cuanto a Fare (coste del pasaje) es, a priori, extraño que ciertos billetes no tuviesen coste alguno, y, aunque podrían ser valores correctos (por ejemplo, podrían ser invitaciones al viaje inaugural), se estudiarán a continuación con más detalle.

Tratamiento

Valores vacíos

En el caso de la variable Cabin, el porcentaje de valores vacíos es demasiado alto (77.46 %) como para poder extrapolarlo o que resulte útil en el estudio, por lo que se decide descartar esta variable.

```
# Se descarta la variable Cabin
titanicData <- select(titanicData, -c(Cabin))

# Se visualizan los primeros registros para comprobar la eliminación realizada
kable(head(titanicData)) %>% kable_styling(latex_options=c("striped", "scale_down"))
```

Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
No	3	male	22	1	0	A/5 21171	7.2500	S
Si	1	female	38	1	0	PC 17599	71.2833	C
Si	3	female	26	0	0	STON/O2. 3101282	7.9250	S
Si	1	female	35	1	0	113803	53.1000	S
No	3	male	35	0	0	373450	8.0500	S
No	3	male	NA	0	0	330877	8.4583	Q

Para el caso de los dos registros con valores vacíos encontrados en Embarked, se van a sustituir por valores NA y serán tratados posteriormente haciendo uso del algoritmo kNN para su imputación.

```
# Se sustituyen los valores vacíos por el valor NA
titanicData$Embarked[titanicData$Embarked==""] <- NA
```

Valores igual a cero

Revisamos qué registros son los que tienen el importe de Fare a 0.

```
# Se visualizan los casos en los que el valor del billete es igual a cero
filter(titanicData, Fare==0)
```

```
##      Survived Pclass  Sex Age SibSp Parch Ticket Fare Embarked
## 1          No      3 male  36     0     0   LINE    0         S
## 2          No      1 male  40     0     0 112059    0         S
## 3          Si      3 male  25     0     0   LINE    0         S
## 4          No      2 male  NA     0     0 239853    0         S
## 5          No      3 male  19     0     0   LINE    0         S
## 6          No      2 male  NA     0     0 239853    0         S
## 7          No      2 male  NA     0     0 239853    0         S
## 8          No      2 male  NA     0     0 239854    0         S
## 9          No      3 male  49     0     0   LINE    0         S
## 10         No      1 male  NA     0     0 112052    0         S
## 11         No      2 male  NA     0     0 239856    0         S
## 12         No      2 male  NA     0     0 239855    0         S
## 13         No      1 male  39     0     0 112050    0         S
## 14         No      1 male  NA     0     0 112058    0         S
## 15         No      1 male  38     0     0  19972    0         S
## 16         No      1 male  NA     0     0 112051    0         S
## 17         No      1 male  49     0     0 112058    0         S
```

No parece apreciarse ningún patrón o característica específica en las observaciones mostradas: parece poco probable que el valor de 0 sea correcto (que se deba, por ejemplo, a invitaciones al viaje inaugural, puesto que hay varias en tercera clase), por lo que se va a proceder a sustituir estos valores perdidos con valores NA para ser tratados junto con el resto de valores perdidos con el algoritmo kNN.

```
# Se sustituyen los valores vacíos por el valor NA
titanicData$Fare[titanicData$Fare==0] <- NA
```

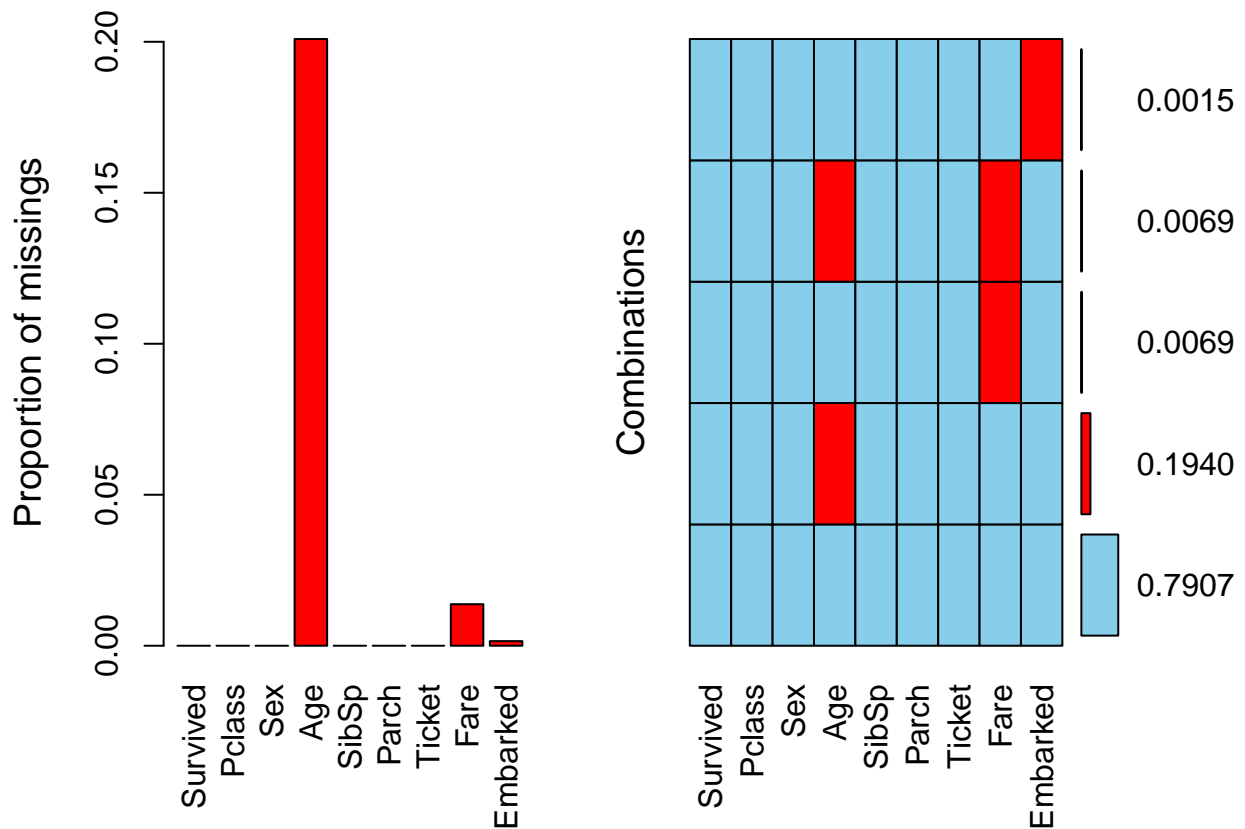
Valores NAs

Para los valores NA existentes en las variables Embarked, Age y Fare, se va a usar la técnica kNN (por sus siglas en inglés, k-Nearest Neighbours), que permite predecir valores en conjuntos de datos multidimensionales formados por datos mixtos (continuos, discretos, ordinales y/o nominales). Este algoritmo encuentra los k vecinos más cercanos y, dependiendo de la variable a imputar, aplica uno de los dos criterios siguientes:

- Si es cualitativa, le asignará el valor más frecuente.
- Si es cuantitativa, le dará el valor de la mediana.

Se comprueba el número de valores NA:

Se comprueba el número de valores NA por variable y se muestran gráficamente
`summary(aggr(titanicData, numbers=TRUE))`



```
##
## Missings per variable:
## Variable Count
## Survived      0
## Pclass        0
## Sex           0
## Age          263
## SibSp         0
## Parch         0
## Ticket        0
## Fare         18
## Embarked      2
##
## Missings in combinations of variables:
```

```
##      Combinations Count      Percent
## 0:0:0:0:0:0:0:0:0 1035 79.0679908
## 0:0:0:0:0:0:0:0:1     2  0.1527884
## 0:0:0:0:0:0:0:1:0     9  0.6875477
## 0:0:0:1:0:0:0:0:0    254 19.4041253
## 0:0:0:1:0:0:0:1:0     9  0.6875477
```

Como puede observarse, la variable con mayor cantidad de valores perdidos es Age, con 263 registros (aproximadamente un 20% del total) sin informar. En la gráfica de la derecha se muestran las combinaciones de variables perdidas. Un 79% de las observaciones tienen todas sus variables informadas, y sólo un 0,7% tienen más de un valor perdido (Fare y Age).

Se procede a realizar la imputación:

```
# Se aplica kNN con el valor de k por defecto, 5, e indicando que no es preciso generar
# las variables informativas de imputación
titanicData <- VIM::kNN(titanicData, imp_var=FALSE)

# Se comprueba que ya no hay valores NA
colSums(is.na(titanicData))
```

```
## Survived   Pclass      Sex      Age      SibSp      Parch      Ticket      Fare
##          0         0         0         0         0         0         0         0
## Embarked
##          0
```

Se vuelven a establecer los niveles de la variable Embarked, ahora ya con sólo tres categorías correspondientes a los tres puertos de embarque:

```
# Se establecen los factores omitiendo el factor NA
titanicData$Embarked <- factor(titanicData$Embarked, levels=c("C","Q","S"))

# Se comprueban los factores y la distribución de valores
levels(titanicData$Embarked)
```

```
## [1] "C" "Q" "S"
```

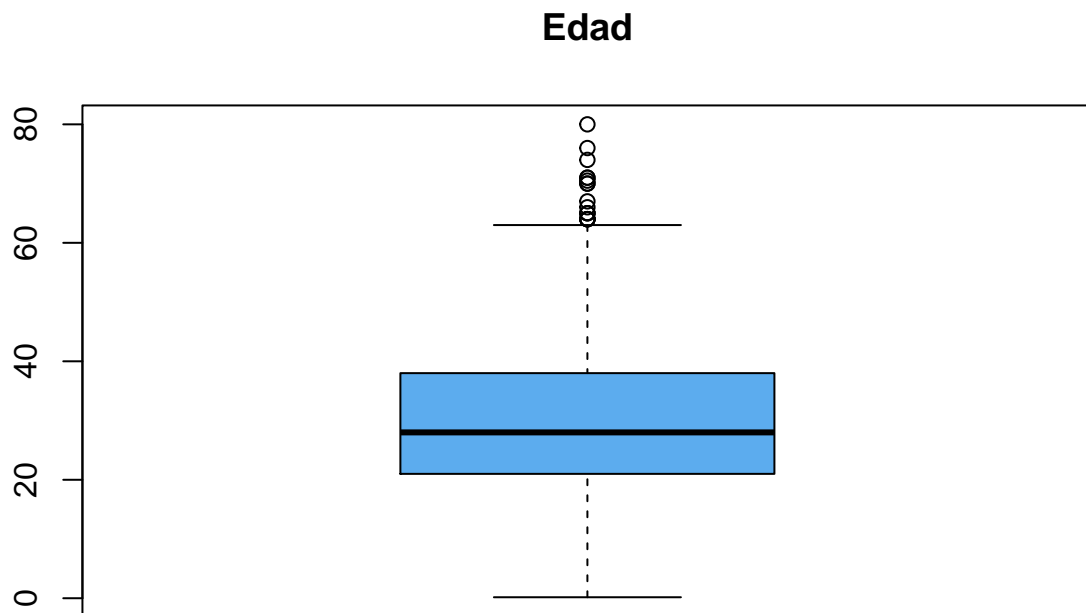
```
summary(titanicData$Embarked)
```

```
##   C   Q   S
## 270 123 916
```

Tratamiento de valores extremos

Procedemos a revisar mediante gráficos de cajas y bigotes posibles valores extremos de las variables.

```
# Boxplot de Age
boxplot(titanicData$Age, main="Edad", col="steelblue2")
```



```
# Mostramos los valores extremos mediante las estadísticas del gráfico
```

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

```
## [1] 66.0 65.0 71.0 70.5 65.0 64.0 65.0 71.0 64.0 80.0 70.0 70.0 74.0 67.0
```

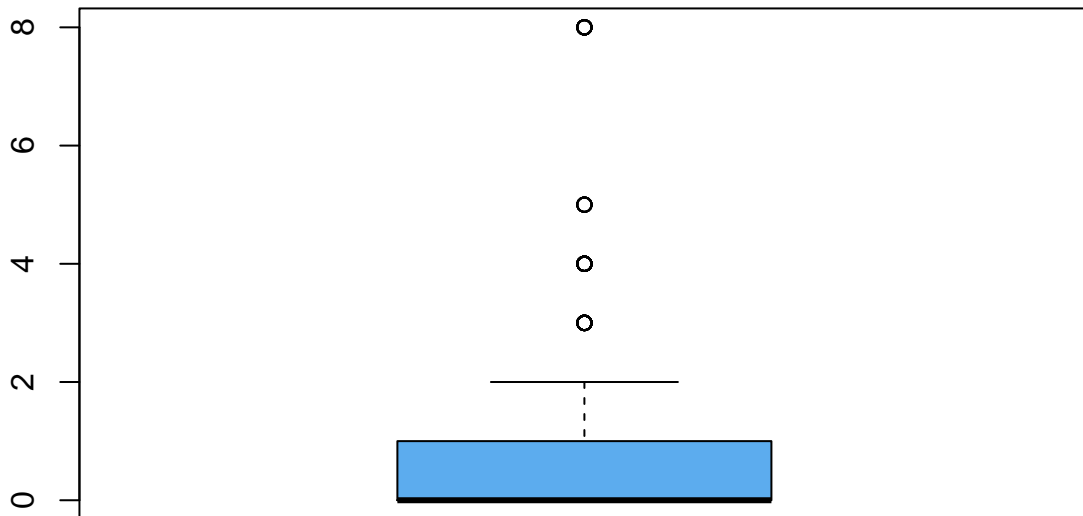
```
## [15] 76.0 64.0 64.0 64.0
```

Dentro de la variable edad se han detectado varios valores considerados extremos pero, tras revisarlos, se comprueba que entran dentro de un rango válido para la edad, por lo que se consideran correctos y no se realizará ningún tratamiento adicional.

```
# Boxplot de SibSp
```

```
boxplot(titanicData$SibSp, main="Número de hermanos/cónyuges", col="steelblue2")
```

Número de hermanos/cónyuges



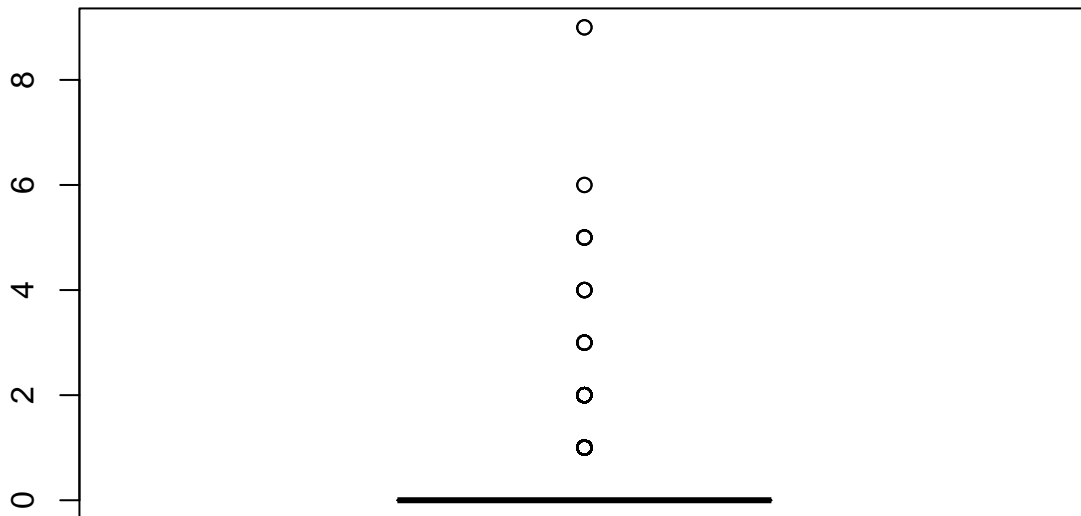
```
# Mostramos los valores extremos
boxplot.stats(titanicData$SibSp)$out
```

```
## [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3
## [36] 5 4 3 4 8 4 3 4 8 4 8 3 4 5 3 4 8 4 8 4 3 3
```

Esta variable recoge el número de hermanos o cónyuges de cada pasajero y, como en el caso anterior, aunque se detectan varios valores extremos, parecen razonables y por tanto no se realizará acción alguna sobre ellos.

```
# Boxplot de Parch
boxplot(titanicData$Parch, main="Número de padres/hijos", col="steelblue2")
```

Número de padres/hijos



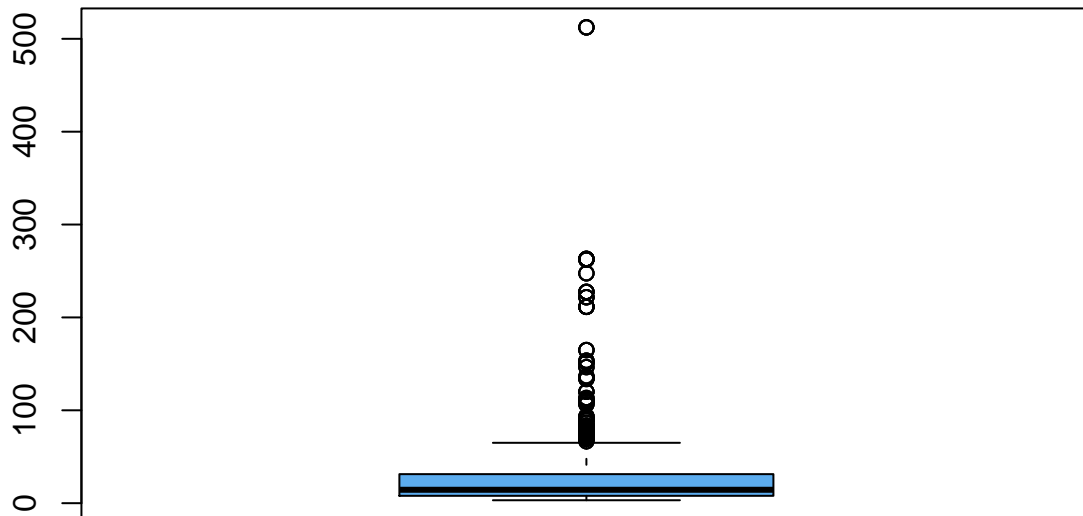
```
# Mostramos los valores extremos
boxplot.stats(titanicData$Parch)$out
```

```
## [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1
## [36] 2 1 1 2 1 4 1 1 1 1 2 2 1 2 1 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 1 1 1 1 1
## [71] 1 2 1 2 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2
## [106] 2 3 4 1 2 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1
## [141] 2 1 1 2 5 2 1 1 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1
## [176] 1 2 1 2 3 1 2 1 2 2 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1
## [211] 1 5 2 1 1 1 1 3 1 2 2 1 2 1 2 1 2 4 1 1 2 1 1 1 4 6 2 3 1 1 2 2 2 1 1
## [246] 2 5 2 3 2 1 1 1 2 1 2 2 2 1 2 1 1 2 1 2 1 2 1 2 2 1 1 1 1 2 1 1 2 1
## [281] 1 1 2 1 2 9 1 1 1 2 2 2 1 9 1 1 2 2 1 1 2 1 1 1 1 1 1 1
```

Esta variable representa el número de hijos o padres. Los valores mostrados también están dentro de un rango razonable y no es posible determinar si alguno de ellos es erróneo, por lo que tampoco se aplicará tratamiento alguno.

```
# Boxplot de Fare
boxplot(titanicData$Fare, main="Tarifa del pasaje", col="steelblue2")
```

Tarifa del pasaje



```
# Mostramos los valores extremos utilizando las estadísticas de $out
boxplot.stats(titanicData$Fare)$out
```

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



```
## [162] 69.5500 69.5500 134.5000 81.8583 262.3750 93.5000 79.2000
## [169] 164.8667 211.5000 90.0000 108.9000
```

Esta variable recoge el precio del pasaje. Sobresale especialmente el importe máximo (superior a 500) por su diferencia con el siguiente valor (alrededor de 200). Mostramos estos registros para intentar deducir si se trata de una errata o su valor es correcto:

```
# Revisamos los registros con mayor valor
kable(filter(titanicData, Fare>500)) %>% kable_styling(latex_options=c("striped", "scale_down"))
```

Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
Si	1	female	35	0	0	PC 17755	512.3292	C
Si	1	male	36	0	1	PC 17755	512.3292	C
Si	1	male	35	0	0	PC 17755	512.3292	C
Si	1	female	58	0	1	PC 17755	512.3292	C

Las cuatro observaciones con el valor máximo corresponden al mismo ticket, por lo que se puede concluir o que ese ticket en concreto fue mal registrado o que es un valor válido y se trata de un pasaje VIP. Tras revisar documentación externa especializada (*Traveler*, 2016) sobre el coste de los billetes en el Titanic y constatar que entran dentro del rango de valores posibles, se determina que son válidos.

Discretización

Ciertos métodos estadísticos requieren u obtienen un mejor desempeño utilizando variables discretas en lugar de continuas. Por ello, se va a proceder a crear nuevas variables que agrupen en distintos grupos los siguientes atributos: Age, Fare, SibSp y Parch.

```
# Discretización para edad
titanicData$AgeD <- cut(titanicData$Age, c(0, 16, 60, 90),
  labels=c("kid", "adult", "elder"),
  include.lowest=TRUE, right=FALSE)

# Discretización para tarifa del pasaje
titanicData$FareD <- cut(titanicData$Fare, c(0, 9, 33, 520),
  labels=c("low", "medium", "high"),
  include.lowest=TRUE, right=FALSE)

# Discretización para número de hermanos/cónyuges
titanicData$SibSpD <- cut(titanicData$SibSp, c(0, 1, 4, 10),
  labels=c("none", "few", "many"),
  include.lowest=TRUE, right=FALSE)

# Discretización para número de padres/hijos
titanicData$ParchD <- cut(titanicData$Parch, c(0, 1, 4, 10),
  labels=c("none", "few", "many"),
  include.lowest=TRUE, right=FALSE)

# Se visualizan algunos datos para comprobar el contenido de las nuevas variables
kable(head(titanicData)) %>% kable_styling(latex_options=c("striped", "scale_down"))
```

Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	AgeD	FareD	SibSpD	ParchD
No	3	male	22	1	0	A/5 21171	7.2500	S	adult	low	few	none
Si	1	female	38	1	0	PC 17599	71.2833	C	adult	high	few	none
Si	3	female	26	0	0	STON/O2. 3101282	7.9250	S	adult	low	none	none
Si	1	female	35	1	0	113803	53.1000	S	adult	high	few	none
No	3	male	35	0	0	373450	8.0500	S	adult	low	none	none
No	3	male	21	0	0	330877	8.4583	Q	adult	low	none	none

```
# Se revisa que se hayan creado como factores
str(titanicData)
```

```
## 'data.frame': 1309 obs. of 13 variables:
## $ Survived: Factor w/ 2 levels "No","Si": 1 2 2 2 1 1 1 1 2 2 ...
## $ Pclass : Factor w/ 3 levels "1","2","3": 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 21 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 ...
## $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
## $ AgeD : Factor w/ 3 levels "kid","adult",...: 2 2 2 2 2 2 2 2 1 2 1 ...
## $ FareD : Factor w/ 3 levels "low","medium",...: 1 3 1 3 1 1 3 2 2 2 ...
## $ SibSpD : Factor w/ 3 levels "none","few","many": 2 2 1 2 1 1 1 2 1 2 ...
## $ ParchD : Factor w/ 3 levels "none","few","many": 1 1 1 1 1 1 1 2 2 1 ...
```

Análisis de los datos

Análisis descriptivo visual

Previo al análisis detallado de los datos y a la elaboración de los tests estadísticos, es importante realizar un primer análisis visual de los mismos para conocer mejor sus características y distribución de valores.

Para las variables cualitativas o categóricas, se utilizarán diagramas de barras o de sectores. Para las variables cuantitativas, además de los diagramas de cajas que ya han sido empleados en el apartado de valores extremos, se emplearán histogramas. Se realizará además un estudio visual sobre posibles relaciones entre variables con el objetivo de eliminar posibles redundancias.

Variable Survived

Esta variable representa si el pasajero sobrevivió (valor “Si”) o no (valor “No”) al hundimiento del Titanic. Es la variable que se pretende explicar o predecir a partir del resto.

Como puede observarse a continuación, el número de pasajeros que fallecieron fue muy elevado, casi dos tercios del total.

```
# Se muestra la tabla de frecuencias
table(titanicData$Survived)
```

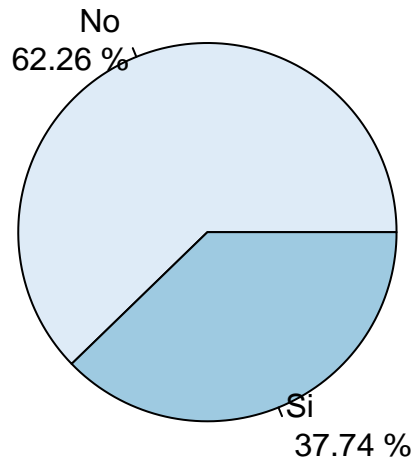
```
##
## No Si
## 815 494
```

```
# Se genera una gráfica de tarta
# Nombres y porcentajes de los sectores
pieNames <- names(table(titanicData$Survived))
piePerc <- round(prop.table(table(titanicData$Survived))*100, 2)
```

```
# Incorporamos a la etiqueta del gráfico los nombres y porcentajes de cada categoría
pieLabels <- paste(pieNames, "\n", piePerc, "%")

pie(table(titanicData$Survived), labels=pieLabels, main="Pasajeros que sobrevivieron",
    col=brewer.pal(2, "Blues"))
```

Pasajeros que sobrevivieron



Variable Pclass

Esta variable representa la clase en la que viaja cada pasajero. Más de la mitad lo hacían en tercera clase, mientras que la proporción entre primera y segunda está más equilibrada.

```
# Se muestra la tabla de frecuencias
pclassTable <- table(titanicData$Pclass)
rownames(pclassTable) <- c("Primera", "Segunda", "Tercera")
pclassTable
```

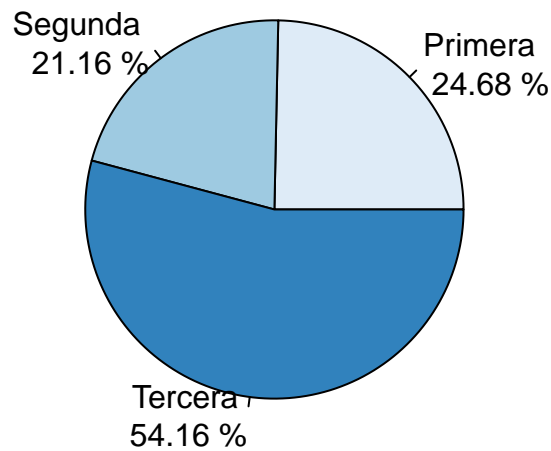
```
##
## Primera Segunda Tercera
##      323      277      709
```

```
# Se genera una gráfica de tarta
# Nombres y porcentajes de los sectores
pieNames <- names(pclassTable)
piePerc <- round(prop.table(pclassTable)*100, 2)
```

```
# Incorporamos a la etiqueta del gráfico los nombres y porcentajes de cada categoría
pieLabels <- paste(pieNames, "\n", piePerc, "%")
```

```
pie(pclassTable, labels=pieLabels, main="Clase del pasaje",
    col=brewer.pal(3, "Blues"))
```

Clase del pasaje



Variable Sex

En cuanto a la distribución por sexos, según puede observarse en el gráfico siguiente, predominaban los hombres, constituyendo casi dos terceras partes de los pasajeros.

```
# Se muestra la tabla de frecuencias
sexTable <- table(titanicData$Sex)
rownames(sexTable) <- c("Mujer", "Hombre")
sexTable
```

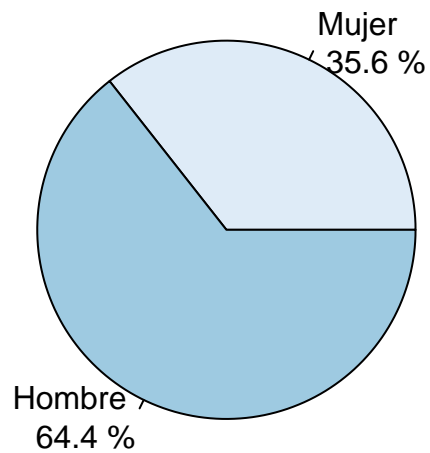
```
##
##  Mujer Hombre
##    466    843
```

```
# Se genera una gráfica de tarta
# Nombres y porcentajes de los sectores
pieNames <- names(sexTable)
piePerc <- round(prop.table(sexTable)*100, 2)
```

```
# Incorporamos a la etiqueta del gráfico los nombres y porcentajes de cada categoría
pieLabels <- paste(pieNames, "\n", piePerc, "%")
```

```
pie(sexTable, labels=pieLabels, main="Sexo",
    col=brewer.pal(3, "Blues"))
```

Sexo



Variable Age

Respecto a la edad, vemos que hay una acumulación de observaciones alrededor de edades intermedias (entre los 20 y 40 años aproximadamente). También puede observarse un sesgo a la derecha, hacia edades superiores.

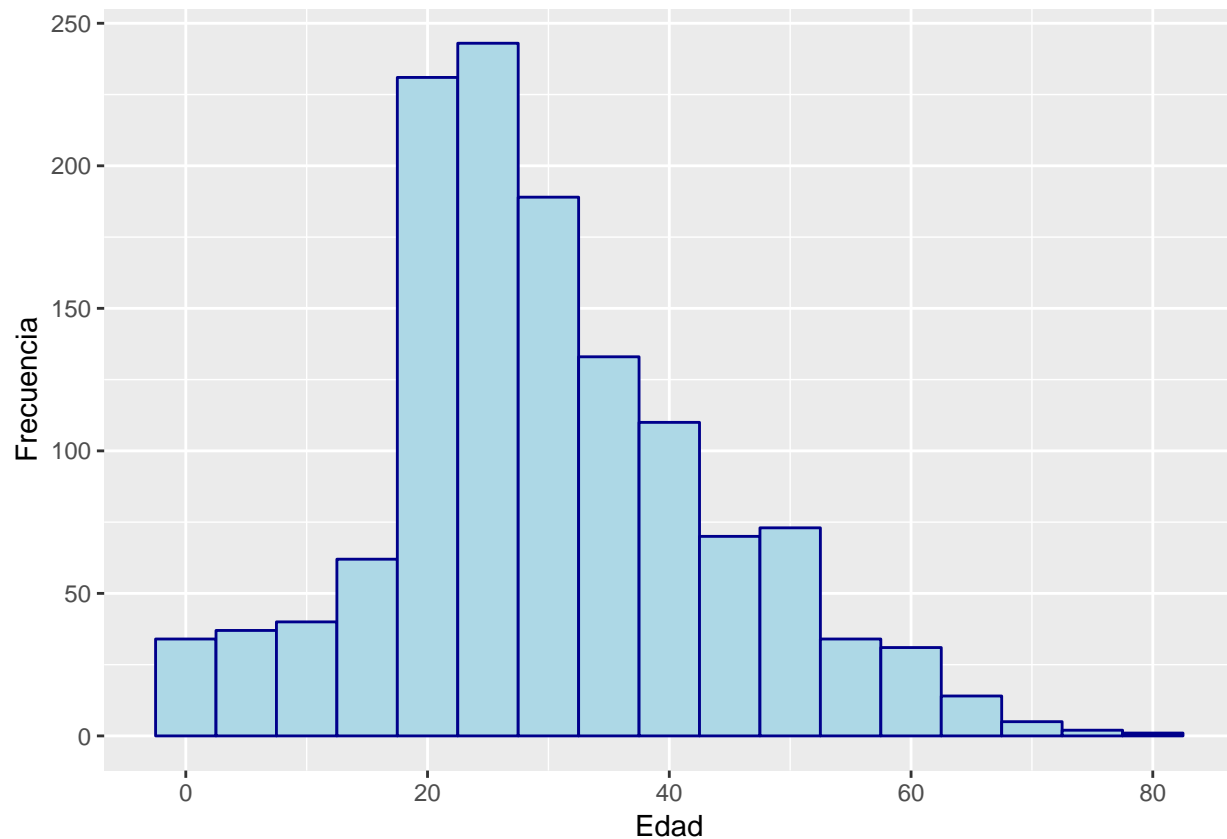
```
# Resumen de los valores de Age
```

```
summary(titanicData$Age)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.17  21.00   28.00   29.79  38.00   80.00
```

```
# Se genera el histograma, estableciendo un ancho de barras de 5 años
```

```
ggplot(data=titanicData, aes(x=Age, fill=Age)) +  
  geom_histogram(binwidth=5, color="darkblue", fill="lightblue") +  
  labs(x="Edad", y="Frecuencia")
```



Variable SibSp

Esta variable indica el número de hermanos y/o cónyuges de los pasajeros. Puede observarse en el gráfico una proporción elevada de personas que viajan sin hermanos ni esposos y el valor atípico que ya habíamos detectado previamente de 8, claramente separado del resto.

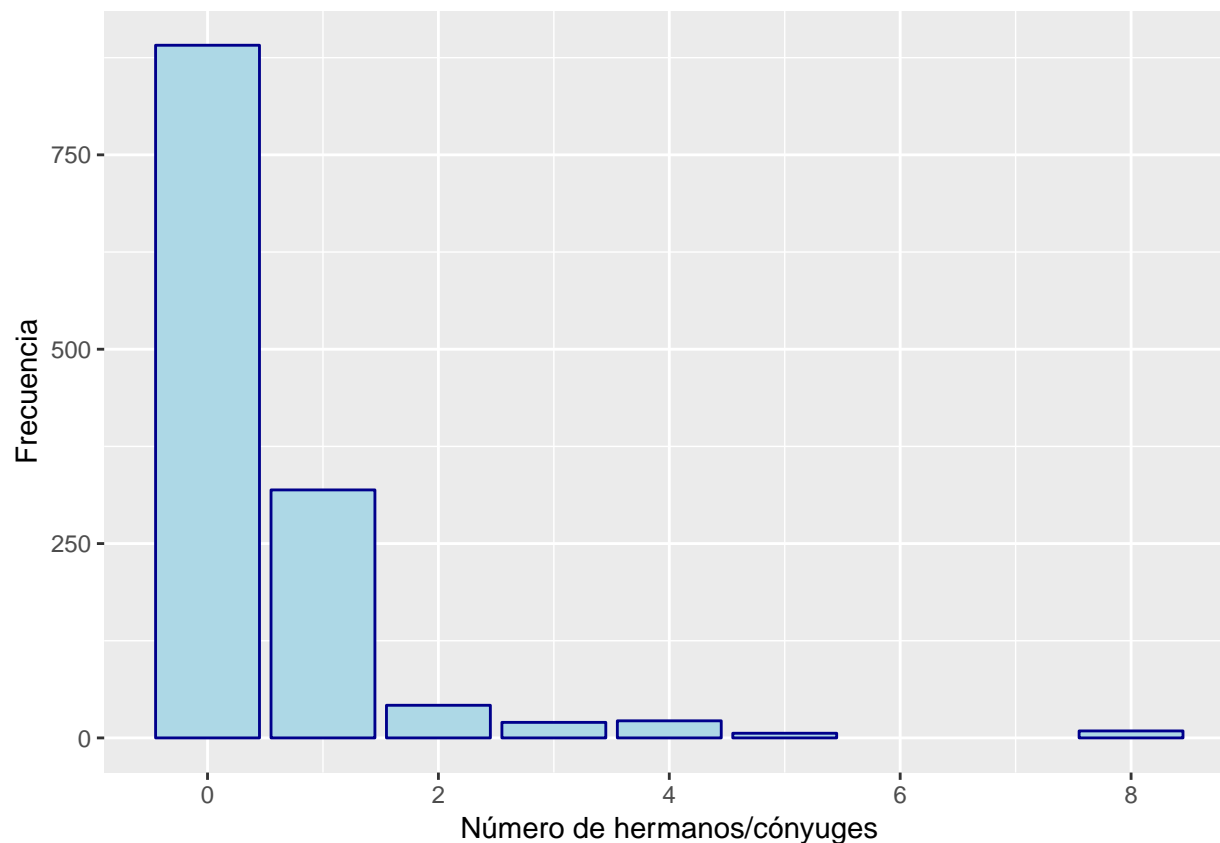
Resumen de los valores de SibSp

```
summary(titanicData$SibSp)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000  0.0000  0.0000  0.4989  1.0000  8.0000
```

Se genera el gráfico de barras

```
ggplot(data=titanicData, aes(x=SibSp, fill=SibSp)) +
  geom_bar(color="darkblue", fill="lightblue") +
  labs(x="Número de hermanos/cónyuges", y="Frecuencia")
```



Variable Parch

Con esta variable se indica el número de padres e hijos de los pasajeros. Puede apreciarse, de manera análoga al caso anterior, que también hay una gran proporción de viajeros que viajaban solos (sin ascendencia/descendencia). También se observa el valor extremo de 9 aislado a la derecha de la grafica.

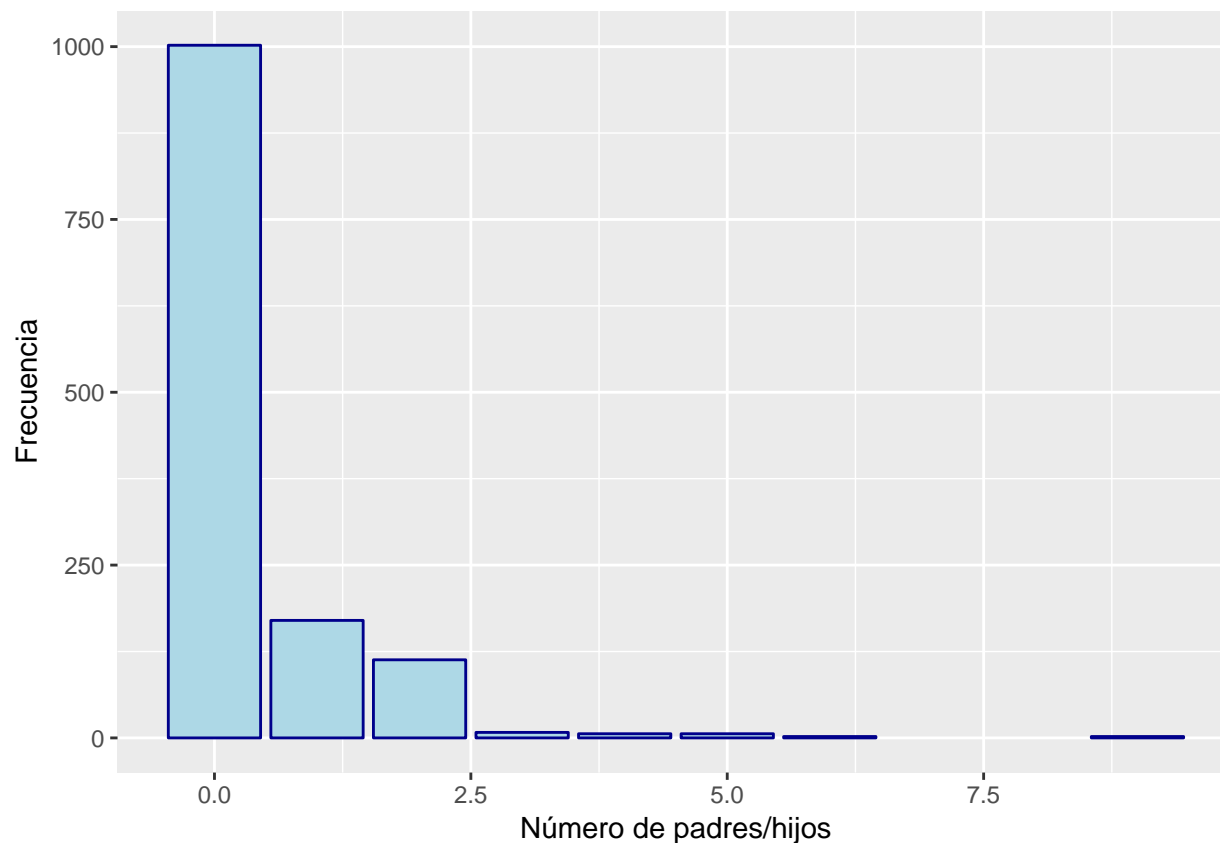
Resumen de los valores de Parch

```
summary(titanicData$Parch)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000  0.000   0.000  0.385  0.000   9.000
```

Se genera el histograma

```
ggplot(data=titanicData, aes(x=Parch, fill=Parch)) +
  geom_bar(color="darkblue", fill="lightblue") +
  labs(x="Número de padres/hijos", y="Frecuencia")
```



Variable Fare

Esta variable representa el precio del pasaje. Aunque hay un rango de precios muy amplio, un 75 % de los valores se concentra en la franja igual o inferior a 65 (ver boxplot para más detalles).

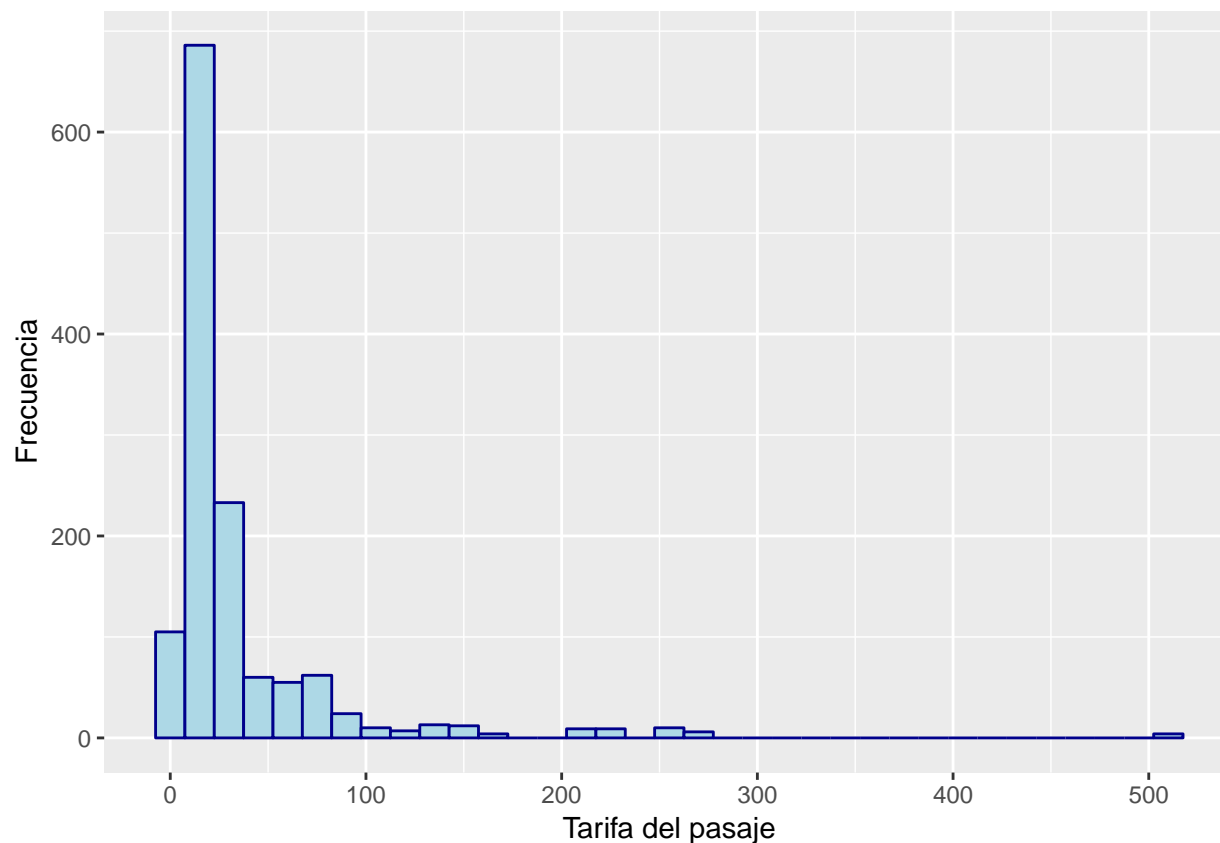
Resumen de los valores de Fare

```
summary(titanicData$Fare)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      3.171   7.925  14.454  33.503  31.275 512.329
```

Se genera el histograma, estableciendo un ancho de barras de 15

```
ggplot(data=titanicData, aes(x=Fare, fill=Fare)) +
  geom_bar(binwidth=15, color="darkblue", fill="lightblue") +
  labs(x="Tarifa del pasaje", y="Frecuencia")
```

Variable Embarked

Esta variable identifica los tres puertos desde los que embarcaron los pasajeros. Según se aprecia en el gráfico de sectores, fue en Southampton donde se recogió la mayor parte de pasajeros.

```
# Se muestra la tabla de frecuencias
embarkedTable <- table(titanicData$Embarked)
rownames(embarkedTable) <- c("Cherbourg", "Queenstown", "Southampton")
embarkedTable
```

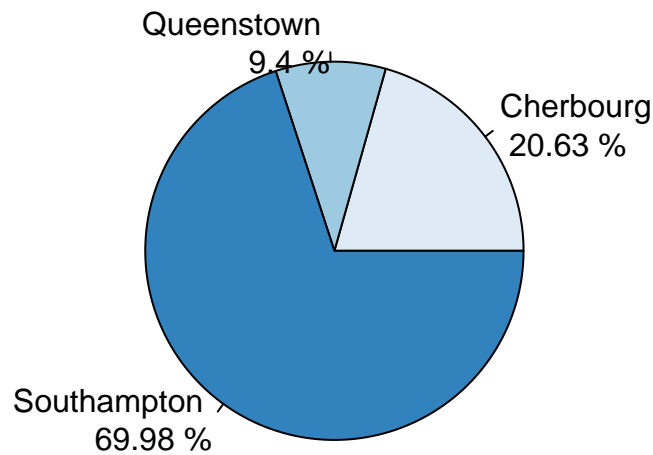
```
##
##   Cherbourg  Queenstown Southampton
##         270         123         916
```

```
# Se genera una gráfica de tarta
# Nombres y porcentajes de los sectores
pieNames <- names(embarkedTable)
piePerc <- round(prop.table(embarkedTable)*100, 2)

# Incorporamos a la etiqueta del gráfico los nombres y porcentajes de cada categoría
pieLabels <- paste(pieNames, "\n", piePerc, "%")

pie(embarkedTable, labels=pieLabels, main="Puerto de embarque",
    col=brewer.pal(3, "Blues"))
```

Puerto de embarque



Correlación entre variables cuantitativas

Dada la influencia que las posibles relaciones entre variables pueden tener en la aplicación de tests estadísticos y la generación del modelo predictivo, vamos a realizar una matriz de correlación entre las variables numéricas existentes en el dataset.

El coeficiente de correlación de Pearson es uno de los más utilizados para identificar relaciones lineales entre variables, pero requiere que estas sigan una distribución normal y que sus varianzas sean iguales (homocedasticidad).

La visualización previa de estas variables mediante histogramas nos hace sospechar que no siguen una distribución normal, pero vamos a utilizar el test de **Shapiro-Wilk** para comprobarlo. En este test la hipótesis nula es que la variable sigue una distribución normal, frente a la hipótesis alternativa en la que la distribución no es normal.

```
# Tests sobre las variables numéricas
shapiro.test(titanicData$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanicData$Age
## W = 0.97621, p-value = 7.132e-14
```

```
shapiro.test(titanicData$SibSp)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  titanicData$SibSp
## W = 0.51108, p-value < 2.2e-16
```

```
shapiro.test(titanicData$Parch)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanicData$Parch
## W = 0.49797, p-value < 2.2e-16
```

```
shapiro.test(titanicData$Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanicData$Fare
## W = 0.52322, p-value < 2.2e-16
```

En todos los tests realizados el p-valor es inferior a 0.05 (nivel de significación por defecto), por lo que no hay evidencia para aceptar la hipótesis nula. En consecuencia, asumimos que las variables **no siguen una distribución normal**. Por tanto, pasamos a aplicar el método de Spearman, test no paramétrico para obtener el grado de dependencia entre variables en las que no se dan los supuestos de normalidad y/o homocedasticidad.

```
# Se buscan posibles correlaciones entre variables numéricas
```

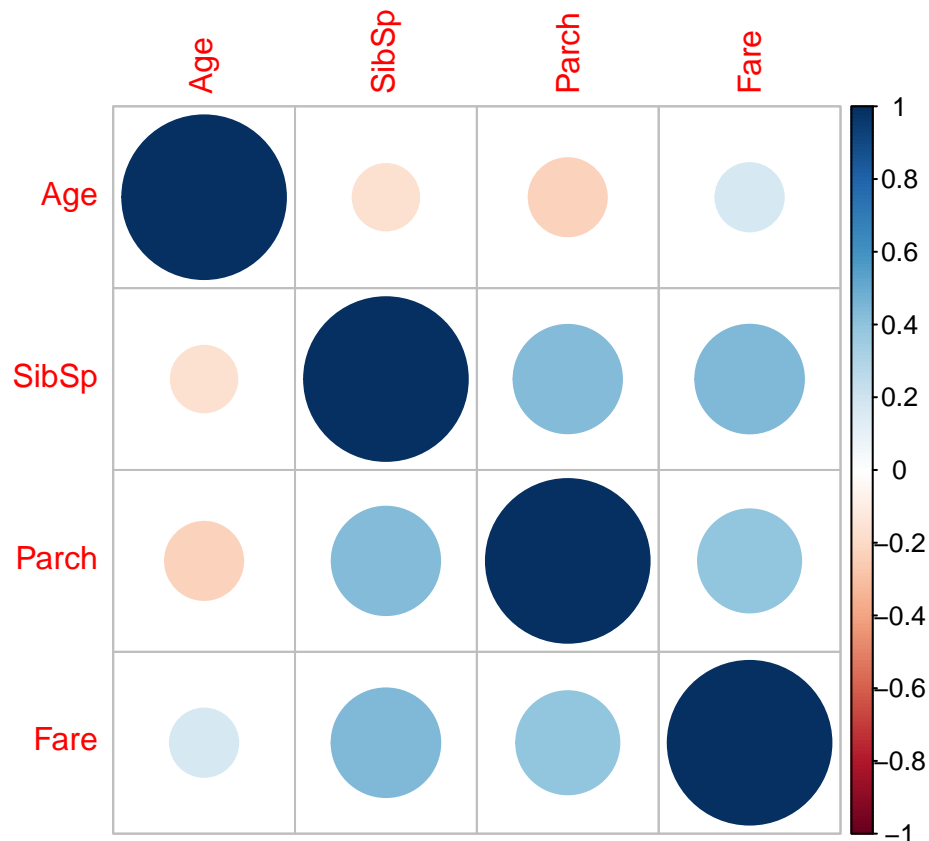
```
numData <- select(titanicData, c(Age, SibSp, Parch, Fare))
```

```
# Se calcula la matriz de correlación y se representa gráficamente
```

```
corMatrix <- cor(numData, method="spearman")
corMatrix
```

```
##           Age      SibSp      Parch      Fare
## Age      1.0000000 -0.1649840 -0.2276328 0.1740247
## SibSp    -0.1649840  1.0000000  0.4383730 0.4405772
## Parch    -0.2276328  0.4383730  1.0000000 0.3967384
## Fare      0.1740247  0.4405772  0.3967384 1.0000000
```

```
corrplot(corMatrix)
```



Puede observarse cierta correlación entre las variables SibSp y Parch, así como entre SibSp y Fare, pero no en un grado que permita descartar inicialmente ninguna de ellas.

Relación de Survived con otras variables

A continuación comprobamos gráficamente si alguna de las variables explicativas cualitativas presentan algún tipo de relación con la variable dependiente. Esta información puede ayudarnos en la selección de variables para los modelos a construir en el apartado de análisis.

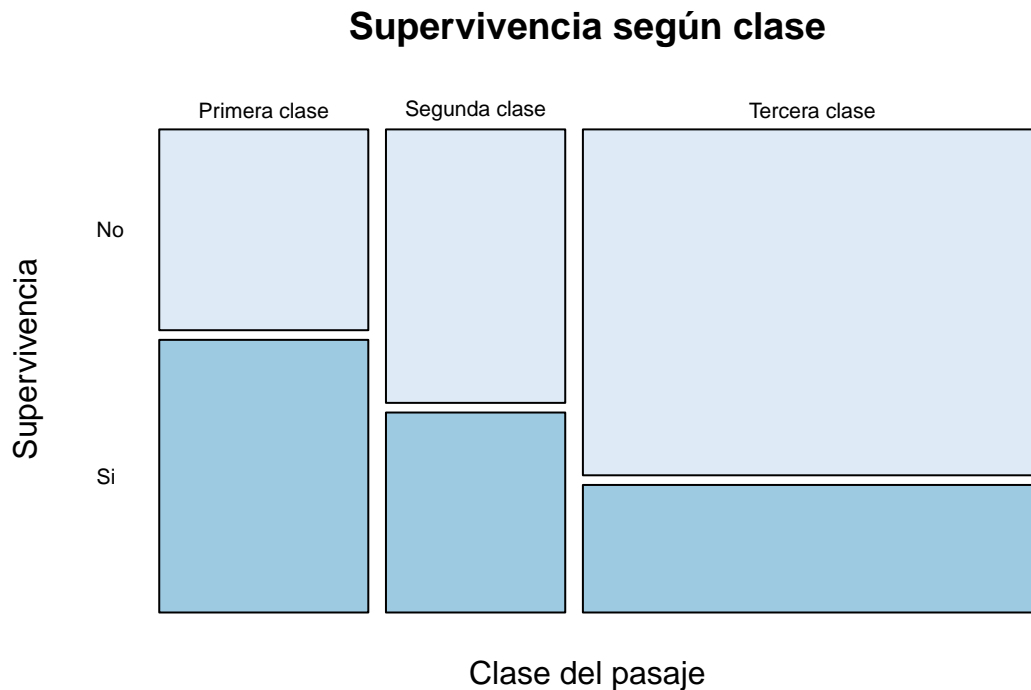
```
# Tabla de contingencia
pclassTable <- table(titanicData$Pclass, titanicData$Survived)
rownames(pclassTable) <- c("Primera clase", "Segunda clase", "Tercera clase")
pclassTable
```

```
##
##           No  Si
## Primera clase 137 186
## Segunda clase 160 117
## Tercera clase 518 191
```

```
# Mostramos las proporciones (totalizando por fila)
prop.table(pclassTable, 1)
```

```
##
##           No      Si
## Primera clase 0.4241486 0.5758514
## Segunda clase 0.5776173 0.4223827
## Tercera clase 0.7306065 0.2693935
```

```
plot(pclassTable, col=brewer.pal(2,"Blues"), main="Supervivencia según clase",
     ylab="Supervivencia", xlab="Clase del pasaje", las=1)
```



Tanto en las tablas de contingencia como en la gráfica, puede observarse claramente que, a pesar de que el número de pasajeros de tercera clase era mayor, en proporción sobrevivieron muchos más pasajeros de primera. De hecho, el porcentaje de fallecidos en tercera es del 73 % aproximadamente, frente al 42 % de primera.

```
# Tabla de contingencia
sexTable <- table(titanicData$Sex, titanicData$Survived)
rownames(sexTable) <- c("Mujer", "Hombre")
sexTable
```

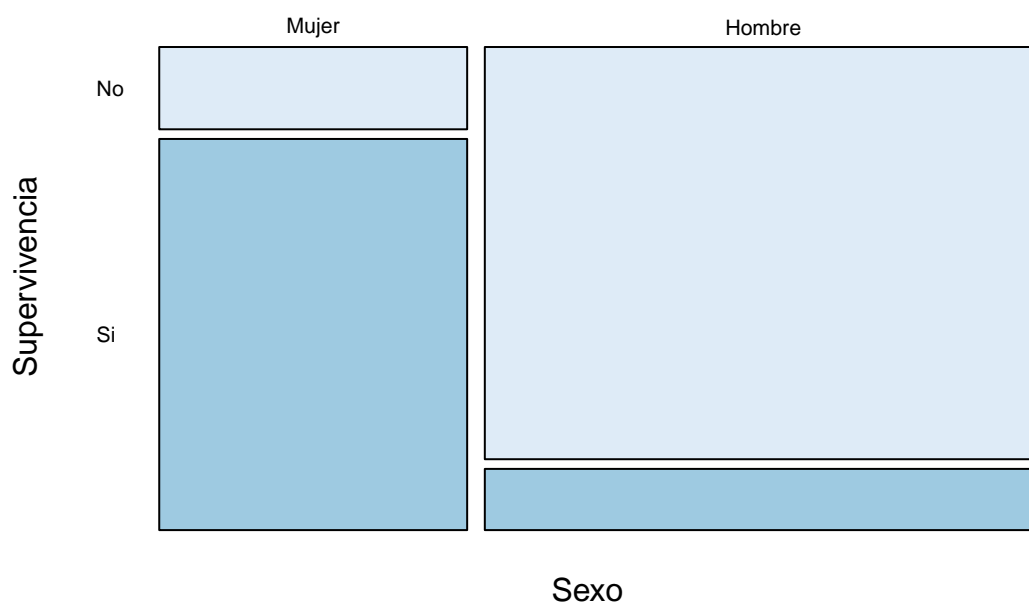
```
##
##           No  Si
##  Mujer    81 385
##  Hombre  734 109
```

```
# Mostramos las proporciones (totalizando por fila)
prop.table(sexTable, 1)
```

```
##
##           No      Si
##  Mujer 0.1738197 0.8261803
##  Hombre 0.8706999 0.1293001
```

```
plot(sexTable, col=brewer.pal(2,"Blues"), main="Supervivencia según sexo",
     ylab="Supervivencia", xlab="Sexo", las=1)
```

Supervivencia según sexo



Aunque el número de hombres era superior entre los pasajeros del Titanic, puede verse que la supervivencia fue mucho menor: casi un 83 % de mujeres se salvaron frente a un 13 % de hombres.

```
# Tabla de contingencia
embarkedTable <- table(titanicData$Embarked, titanicData$Survived)
rownames(embarkedTable) <- c("Cherbourg", "Queenstown", "Southampton")
embarkedTable
```

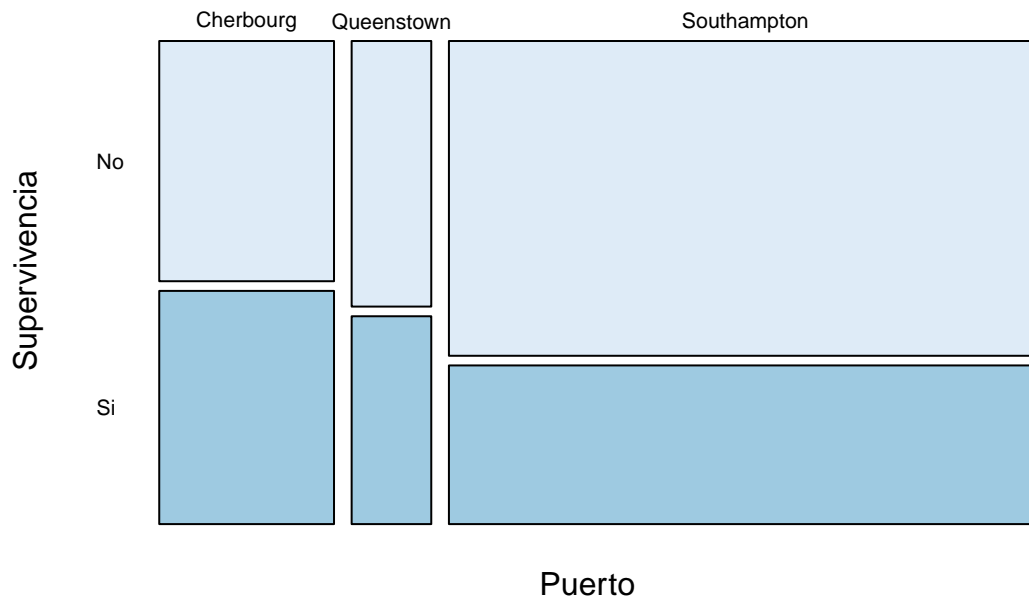
```
##
##           No  Si
## Cherbourg  137 133
## Queenstown   69  54
## Southampton 609 307
```

```
# Mostramos las proporciones (totalizando por fila)
prop.table(embarkedTable, 1)
```

```
##
##           No      Si
## Cherbourg 0.5074074 0.4925926
## Queenstown 0.5609756 0.4390244
## Southampton 0.6648472 0.3351528
```

```
plot(embarkedTable, col=brewer.pal(2,"Blues"),
     main="Supervivencia según puerto de embarque",
     ylab="Supervivencia", xlab="Puerto", las=1)
```

Supervivencia según puerto de embarque



En cuanto al puerto de embarque, se aprecian ciertas diferencias, siendo más relevantes en el caso de Southampton, donde el índice de supervivencia es perceptiblemente menor respecto al siguiente de los puertos (más de 10 puntos porcentuales). Quizá esta discrepancia se deba a otras características comunes que pudieran tener los pasajeros de un puerto u otro, por lo que vamos a investigar más detalladamente la relación de esta variable con otras.

Relación entre variables categóricas

Revisamos la relación entre la clase y el puerto de embarque:

```
# Tabla de contingencia
pclassEmbarkedTable <- table(titanicData$Embarked, titanicData$Pclass)
colnames(pclassEmbarkedTable) <- c("Primera clase", "Segunda clase", "Tercera clase")
rownames(pclassEmbarkedTable) <- c("Cherbourg", "Queenstown", "Southampton")
pclassEmbarkedTable
```

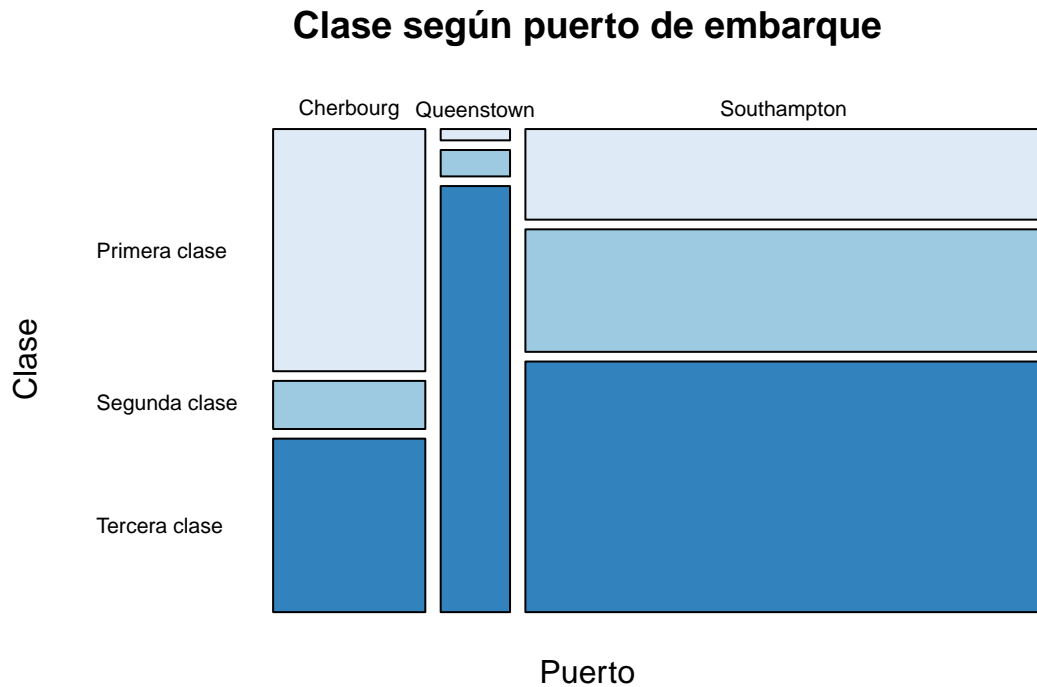
```
##
##      Primera clase Segunda clase Tercera clase
## Cherbourg      141          28          101
## Queenstown       3           7          113
## Southampton     179         242          495
```

```
# Mostramos las proporciones (totalizando por fila)
prop.table(pclassEmbarkedTable, 1)
```

```
##
##      Primera clase Segunda clase Tercera clase
## Cherbourg      0.5222222  0.1037037  0.37407407
```

```
## Queenstown 0.02439024 0.05691057 0.91869919
## Southampton 0.19541485 0.26419214 0.54039301
```

```
plot(pclassEmbarkedTable, col=brewer.pal(2,"Blues"),
     main="Clase según puerto de embarque",
     ylab="Clase", xlab="Puerto", las=1)
```



Como puede observarse, la proporción de pasajeros de primera clase que embarcaron en Cherbourg es mucho mayor que los de Southampton (52.2 % vs 19.5 %).

```
# Tabla de contingencia
pclassEmbarkedFTable <- table(titanicData$Embarked, titanicData$FareD)
colnames(pclassEmbarkedFTable) <- c("Bajo", "Medio", "Alto")
rownames(pclassEmbarkedFTable) <- c("Cherbourg", "Queenstown", "Southampton")
pclassEmbarkedFTable
```

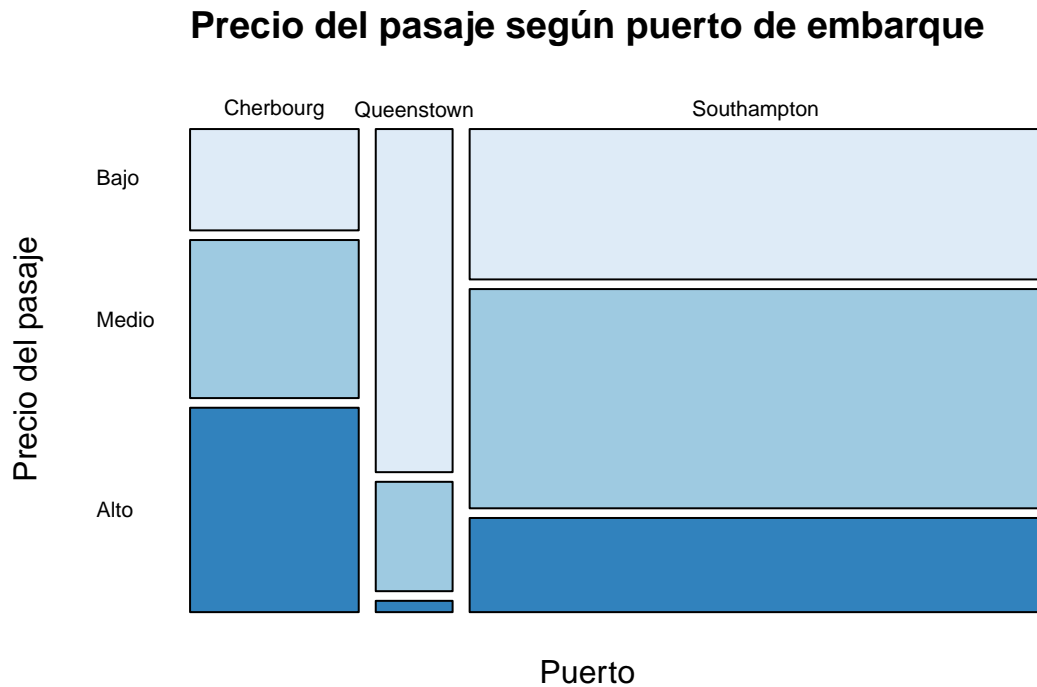
```
##
##           Bajo Medio Alto
## Cherbourg    59   92  119
## Queenstown   91   29   3
## Southampton  297  433  186
```

```
# Mostramos las proporciones (totalizando por fila)
prop.table(pclassEmbarkedFTable, 1)
```

```
##
##           Bajo      Medio      Alto
## Cherbourg  0.21851852 0.34074074 0.44074074
```



```
## Queenstown 0.73983740 0.23577236 0.02439024
## Southampton 0.32423581 0.47270742 0.20305677
plot(pclassEmbarkedFTable, col=brewer.pal(2,"Blues"),
     main="Precio del pasaje según puerto de embarque",
     ylab="Precio del pasaje", xlab="Puerto", las=1)
```



En Cherbourg más de un 40 % de los billetes tenían una tarifa alta, en contraste con Southampton, donde predominaban los de precio medio y bajo, y Queenstown, donde alrededor del 74 % de los billetes eran de coste bajo.

Revisamos ahora la posible relación entre la clase y el precio del billete:

```
# Tabla de contingencia
pclassFareTable <- table(titanicData$Pclass, titanicData$FareD)
colnames(pclassFareTable) <- c("Bajo", "Medio", "Alto")
rownames(pclassFareTable) <- c("Primera clase", "Segunda clase", "Tercera clase")
pclassFareTable
```

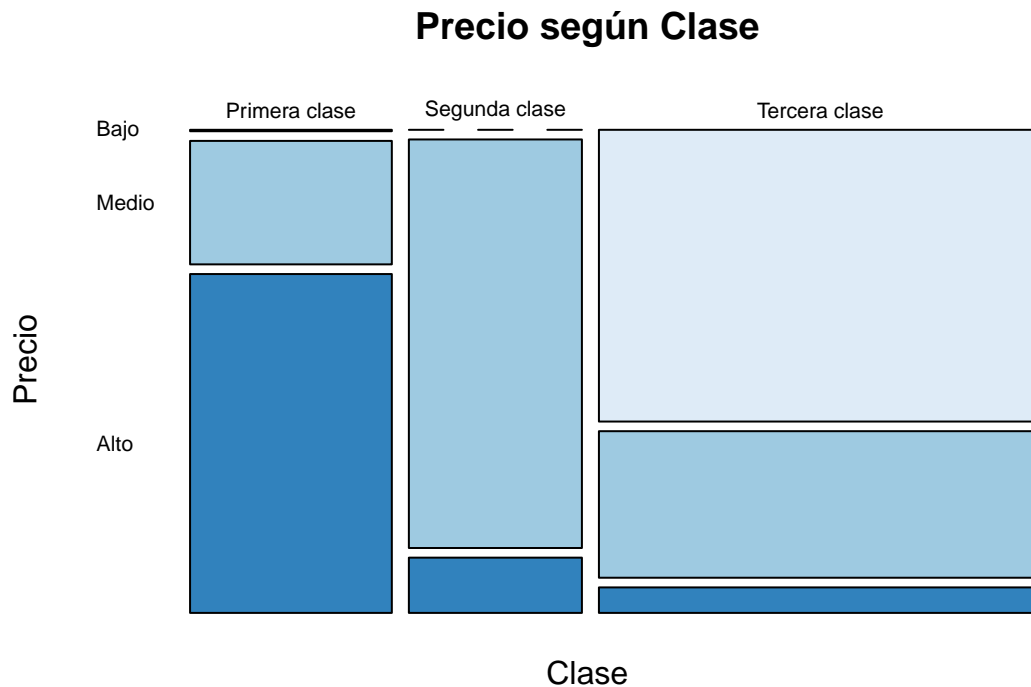
```
##
##           Bajo Medio Alto
## Primera clase     1   86 236
## Segunda clase     0  244  33
## Tercera clase  446  224  39
```

```
# Mostramos las proporciones (totalizando por fila)
prop.table(pclassFareTable, 1)
```

```
##
```

```
##           Bajo      Medio      Alto
## Primera clase 0.003095975 0.266253870 0.730650155
## Segunda clase 0.000000000 0.880866426 0.119133574
## Tercera clase 0.629055007 0.315937941 0.055007052
```

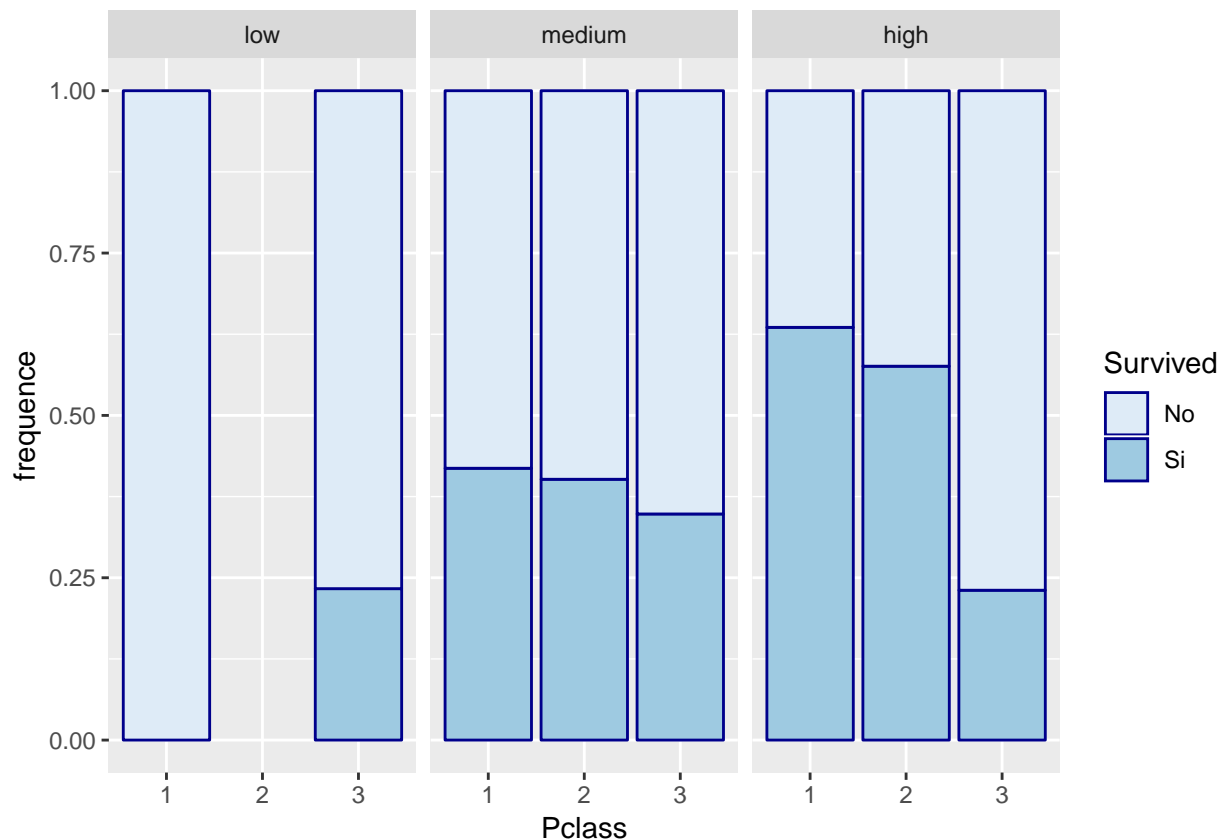
```
plot(pclassFareTable, col=brewer.pal(2,"Blues"),
     main="Precio según Clase",
     ylab="Precio", xlab="Clase", las=1)
```



Como era de esperar, la mayor proporción de billetes caros se da en primera clase, los de coste medio están sobre todo ubicados en segunda clase y, por último, los de tercera suelen ser los más baratos.

Por último, se representa la relación entre las variables de clase, precio del billete y supervivencia:

```
# Se visualiza la relación entre Pclass, FareD y Survived
ggplot(data=titanicData[1:length(titanicData$Survived),], aes(x=Pclass, fill=Survived)) +
  geom_bar(position="fill", color="darkblue") + facet_wrap(~FareD) + ylab("frequence") +
  scale_fill_brewer(palette="Blues")
```



Esta gráfica parece sugerir que, además de la clase, el precio del billete también es relevante, sobre todo en primera y segunda clase, donde el número de supervivientes es mayor si el coste del billete es superior.

Planificación

El análisis que se va a llevar a cabo pretende establecer un modelo de predicción de la variable Survived considerando algunas de las características de los pasajeros del Titanic. Para ello, se ha decidido evaluar tres métodos de predicción distintos:

- Árbol de decisión: se utilizará el algoritmo C5.0. Para aquellas variables numéricas como edad o precio del pasaje, se utilizarán las variables discretizadas que preparamos en apartados anteriores.
- Regresión logística: que nos permite predecir la probabilidad de ocurrencia de una variable dependiente dicotómica (Survived) a partir de la influencia, indicada por unos coeficientes calculados por el algoritmo, de las variables independientes.
- Random Forest: donde se generan varios árboles de decisión simultáneamente incluyendo al azar las variables predictoras.

Para escoger aquella técnica que presenta un mejor desempeño, se realizará una evaluación de todas ellas utilizando una validación cruzada, *k-fold cross validation*. Para ello, es preciso realizar una división del conjunto de datos en *k* subconjuntos e ir alternándolos en el entrenamiento y prueba de los modelos (*k* iteraciones). Las métricas de comparación entre modelos que se emplearán serán la precisión y el coeficiente kappa de Cohen:

- Precisión: es el porcentaje de clasificaciones correctas de todas las instancias.
- Kappa: similar a la precisión, pero ajustando el efecto del azar en la proporción de la concordancia observada.

Comprobación de normalidad y homocedasticidad

En apartados previos ya tuvimos que aplicar el test de Shapiro-Wilk para seleccionar el método utilizar en el estudio de correlación entre variables, y se concluyó que las variables estudiadas no presentaban una distribución normal.

Aunque el estudio de homocedasticidad no es preciso para el objetivo principal que se persigue en este análisis, vamos a proceder, con fines meramente académicos, a analizar las varianzas de las edades en dos grupos: supervivientes frente a fallecidos.

Dado que la variable Age no sigue una distribución normal, aplicamos el test de Levene (utilizando la mediana) para comprobar si los dos grupos tienen igualdad de varianzas.

```
# Se aplica el test de Levene
leveneTest(Age~Survived, data=titanicData, location="median")

## Levene's Test for Homogeneity of Variance (center = median: "median")
##           Df F value Pr(>F)
## group      1  1.1886 0.2758
##           1307
```

El p-valor es superior a 0.05, por lo que **no hay diferencias significativas entre las varianzas de los dos grupos**.

Aplicación de pruebas estadísticas

En primer lugar, se realiza la división del conjunto de datos en 10 subconjuntos. Se ha escogido k=10 por ser un valor que empíricamente ha demostrado buenos resultados en tareas de data mining y machine learning (Brownlee, 2018).

```
# Se utiliza el paquete caret para realizar la evaluación de modelos. Indicamos que se
# va a realizar una validación cruzada con k=10
fitControl <- trainControl(method="cv", number=10, savePredictions=TRUE)
```

Árbol de decisión

A continuación, se procede a evaluar el algoritmo C5.0 de árboles de decisión con k-fold validation:

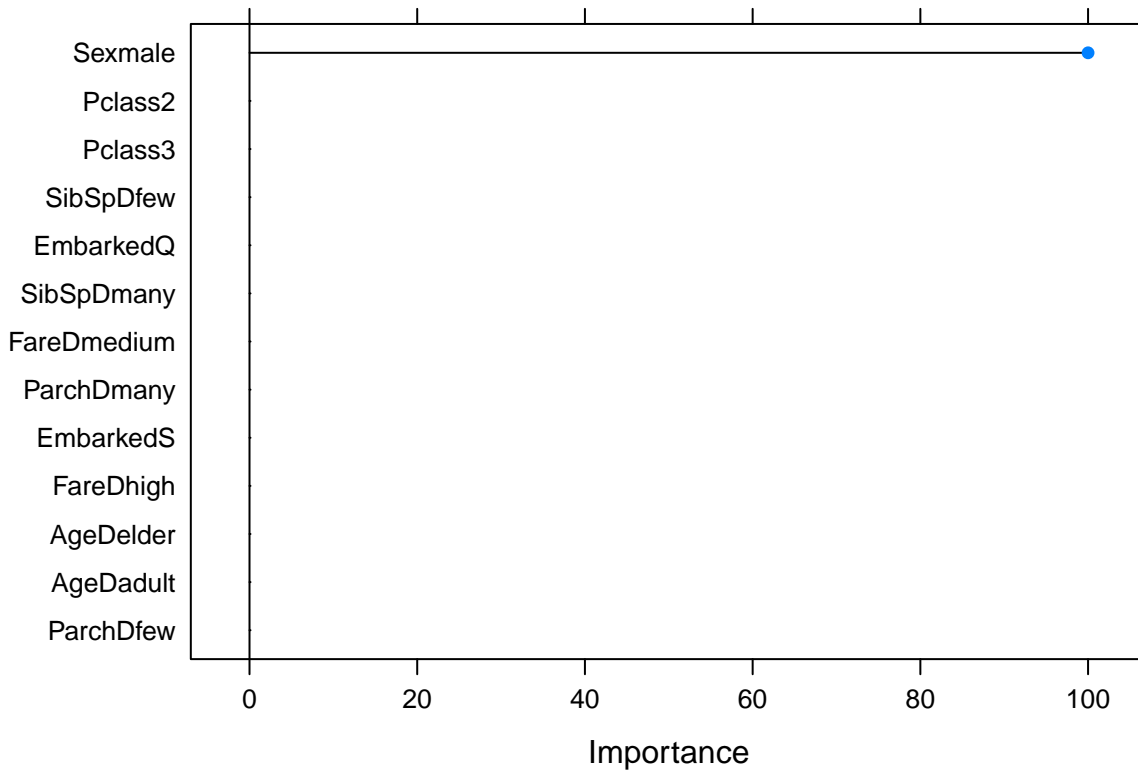
```
## Estimamos la precisión para el algoritmo C5.0
set.seed(0806)
titanicTree1 <- train(Survived ~ Pclass + Sex + AgeD + SibSpD + ParchD + FareD + Embarked,
                      data=titanicData, method="C5.0Tree", trControl=fitControl)

# Mostramos los resultados obtenidos
print(titanicTree1)

## Single C5.0 Tree
##
## 1309 samples
##    7 predictor
##    2 classes: 'No', 'Si'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1177, 1178, 1178, 1178, 1179, 1179, ...
## Resampling results:
##
##   Accuracy   Kappa
```

```
## 0.8549053 0.6877494
```

```
# Así como las variables según su importancia en el modelo construido  
plot(varImp(titanicTree1))
```



Se ha obtenido un modelo con una precisión del 85.49% y un kappa de 0.6877. Además, la única variable predictora que utiliza el modelo es el sexo del pasajero.

Modelo de regresión logística

Con el objetivo de intentar mejorar el desempeño del modelo anterior, utilizaremos un modelo de regresión logística. En él, se evaluará la probabilidad de que un pasajero sobreviva al hundimiento (variable dependiente dicotómica) mediante el uso de ciertas variables explicativas.

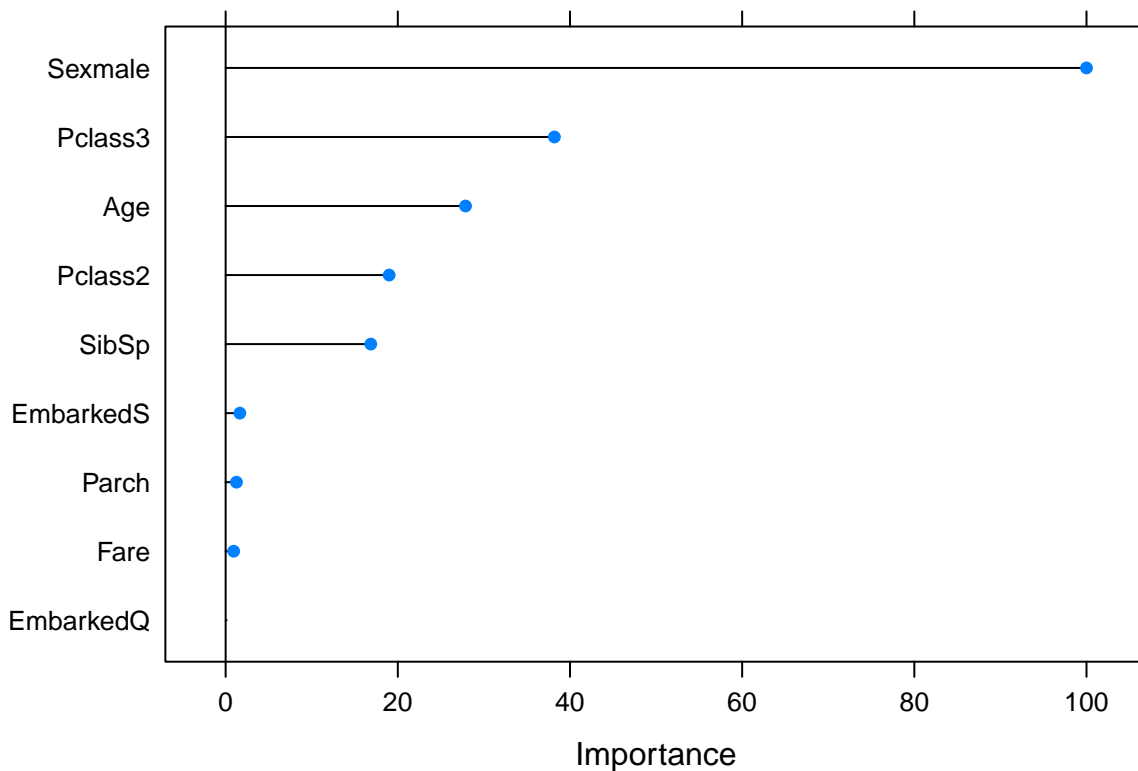
Nuestra primera versión del modelo utiliza como variables predictoras todas las variables originales del dataset (a excepción de las descartadas al comienzo de la práctica).

```
## Estimamos la precisión con el modelo de regresión logística  
set.seed(0806)  
titanicGlm1 <- train(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,  
                     data=titanicData, method="glm", family=binomial(),  
                     trControl=fitControl)
```

```
# Mostramos el resultado del modelo y las variables según su importancia  
print(titanicGlm1)
```

```
## Generalized Linear Model  
##  
## 1309 samples
```

```
##      7 predictor
##      2 classes: 'No', 'Si'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1177, 1178, 1178, 1178, 1179, 1179, ...
## Resampling results:
##
## Accuracy   Kappa
## 0.8495324  0.6758223
plot(varImp(titanicGlm1))
```



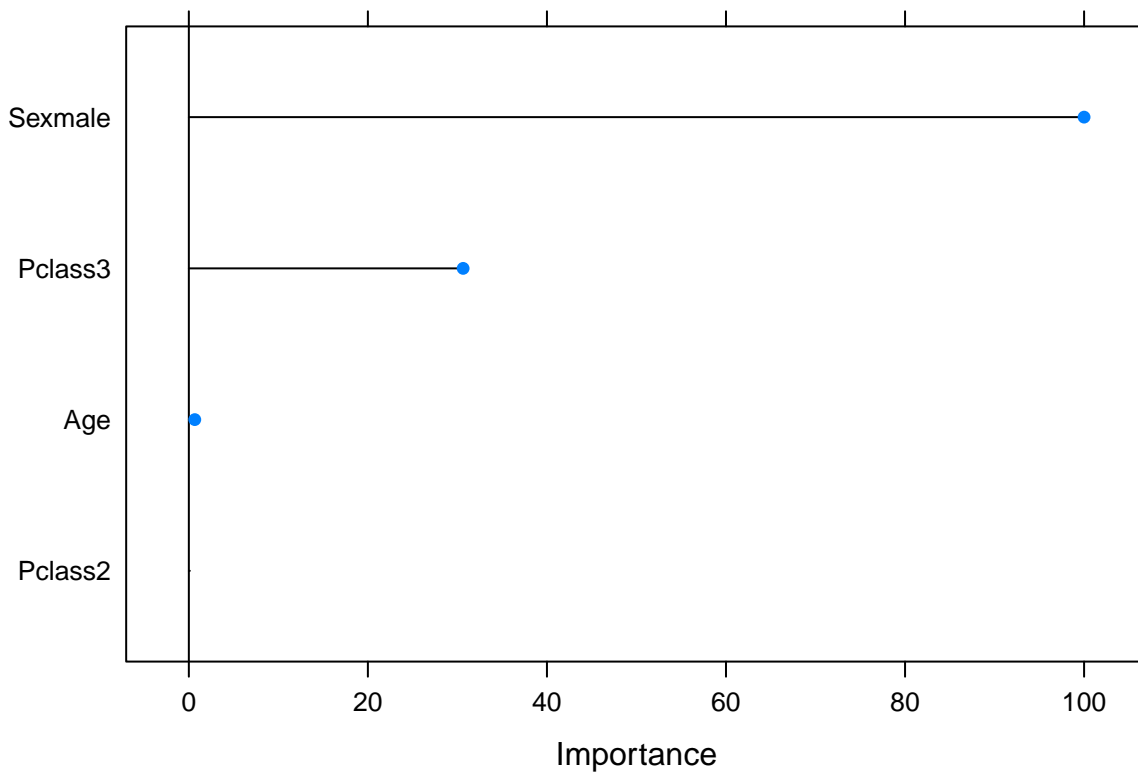
Se ha obtenido un modelo con una precisión del 84.95 % y un kappa de 0.6758. Las variables más importantes en la elaboración del modelo son el sexo del pasajero, la clase y la edad, por lo que vamos a intentar afinar el modelo eliminando las variables con menos peso.

```
## Evaluamos el modelo de regresión logística
set.seed(0806)
titanicGlm2 <- train(Survived ~ Pclass + Sex + Age,
                     data=titanicData, method="glm", family=binomial(),
                     trControl=fitControl)

# Mostramos los resultados y la importancia de las variables
print(titanicGlm2)

## Generalized Linear Model
##
```

```
## 1309 samples
##    3 predictor
##    2 classes: 'No', 'Si'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1177, 1178, 1178, 1178, 1179, 1179, ...
## Resampling results:
##
## Accuracy   Kappa
## 0.8549053  0.6888915
plot(varImp(titanicGlm2))
```



Con este ajuste se ha conseguido una precisión pareja a la del modelo de C5.0 pero se ha mejorado ligeramente el valor de kappa, por lo que, de momento, este es el modelo con mejor desempeño.

Random forest

En este apartado se hace uso del algoritmo predictivo **random forest**, que realiza varias combinaciones de árboles de decisión utilizando distintas observaciones y variables en cada uno de ellos. Para la predicción de la variable contabiliza los resultados de la misma obtenidos en todos los árboles y da por bueno aquel que se repite en más ocasiones.

Realizamos una primera ejecución considerando todas las variables (a excepción de las versiones discretizadas de Age, SibSp, Parch y Fare):

```

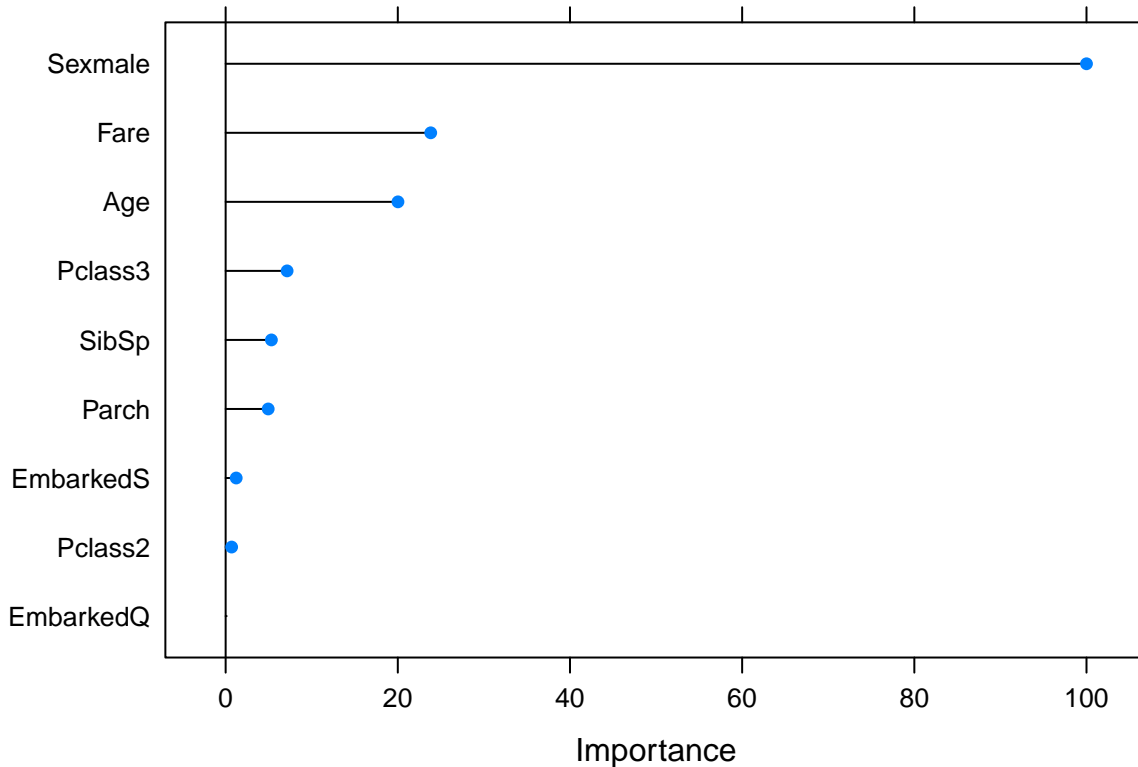
## Evaluamos el modelo de random forest
set.seed(0806)
titanicRf1 <- train(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,
                    data=titanicData, method="rf", trControl=fitControl)

# Mostramos los resultados
print(titanicRf1)

## Random Forest
##
## 1309 samples
##    7 predictor
##    2 classes: 'No', 'Si'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1177, 1178, 1178, 1178, 1179, 1179, ...
## Resampling results across tuning parameters:
##
##  mtry  Accuracy  Kappa
##  2     0.8685994  0.7150449
##  5     0.8533028  0.6843683
##  9     0.8548121  0.6888338
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.

plot(varImp(titanicRf1))

```

Las variables SibSp, Parch y Embarked son las que menor importancia presentan para el modelo, por lo que vamos a repetir la evaluación del modelo eliminándolas:

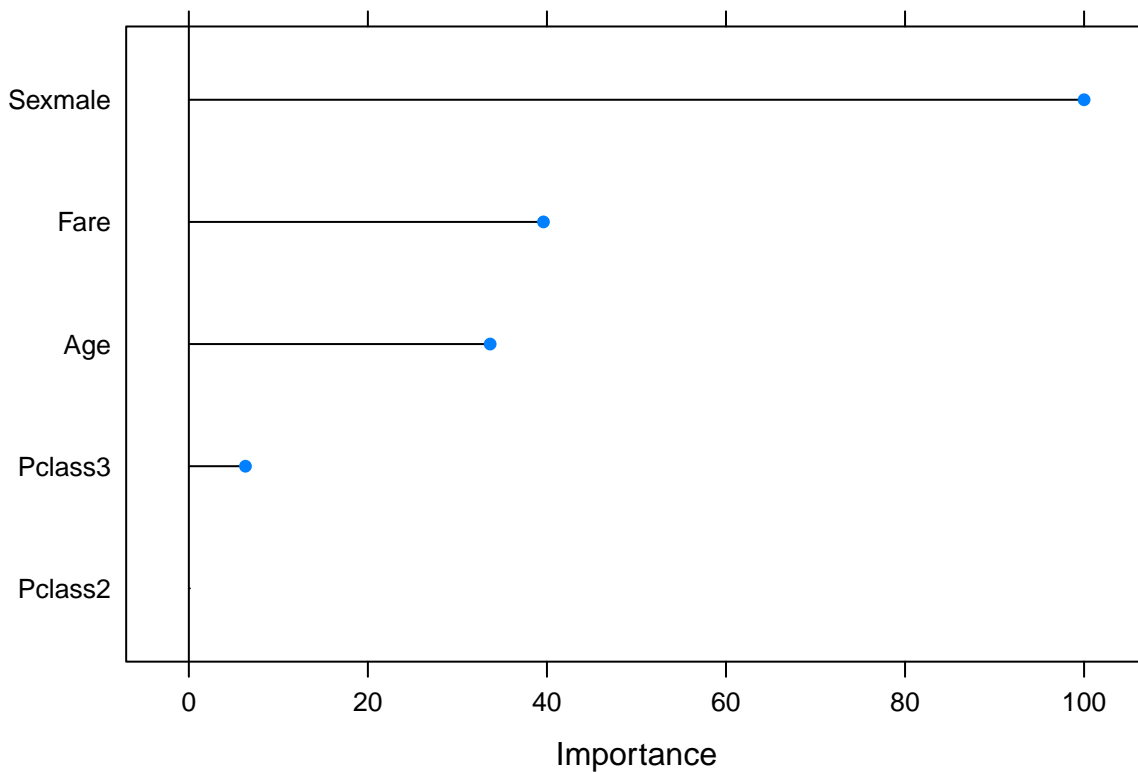
```
## Evaluamos el modelo de random forest
set.seed(0806)
titanicRf2 <- train(Survived ~ Pclass + Sex + Age + Fare,
                    data=titanicData, method="rf", trControl=fitControl)

# Se muestran los resultados obtenidos
print(titanicRf2)
```

```
## Random Forest
##
## 1309 samples
## 4 predictor
## 2 classes: 'No', 'Si'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1177, 1178, 1178, 1178, 1179, 1179, ...
## Resampling results across tuning parameters:
##
## mtry Accuracy Kappa
## 2 0.8670787 0.7121322
## 3 0.8685997 0.7180824
## 5 0.8494684 0.6761783
##
```

```
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

```
plot(varImp(titanicRf2))
```



Como se puede observar, se han mejorado los valores de precisión y kappa de la versión anterior, con unos valores de 86.86 % y 0.7181 respectivamente.

Representación de los resultados

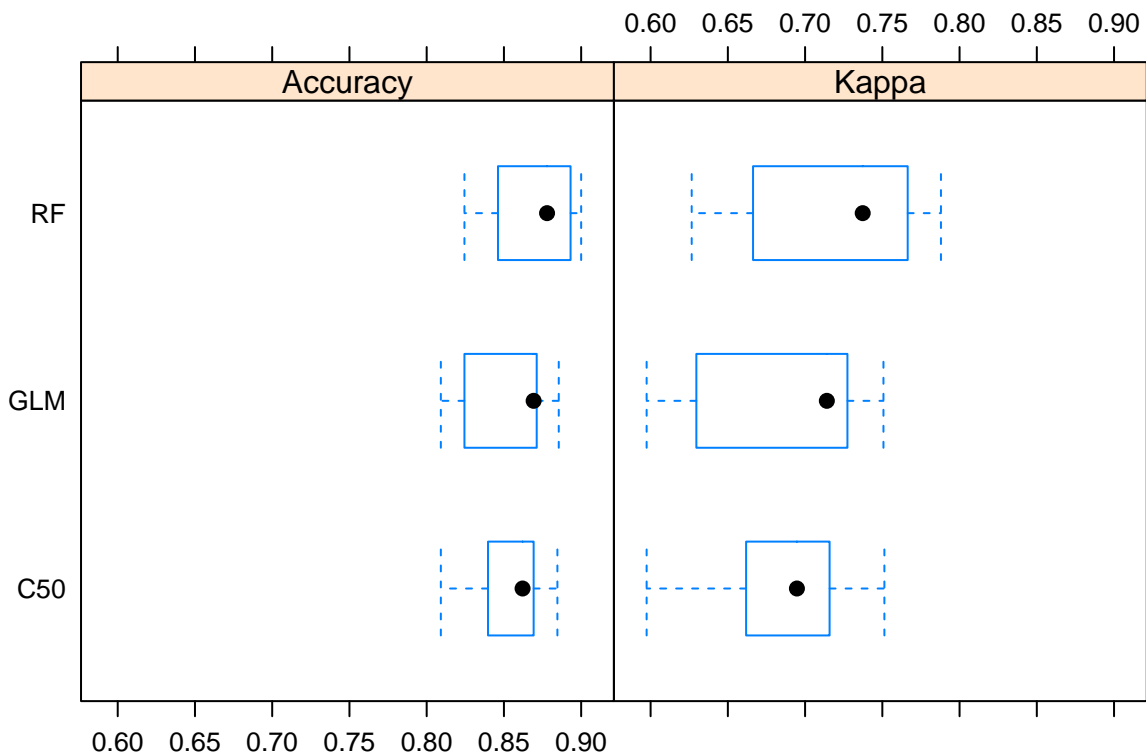
Vamos a proceder a comparar los resultados obtenidos para cada uno de los modelos probados en el apartado anterior; para ello, utilizaremos la función `resamples` del paquete `caret`.

```
# Almacenamos los resultados de los tres modelos
results <- resamples(list(C50=titanicTree1, GLM=titanicGlm2, RF=titanicRf2))

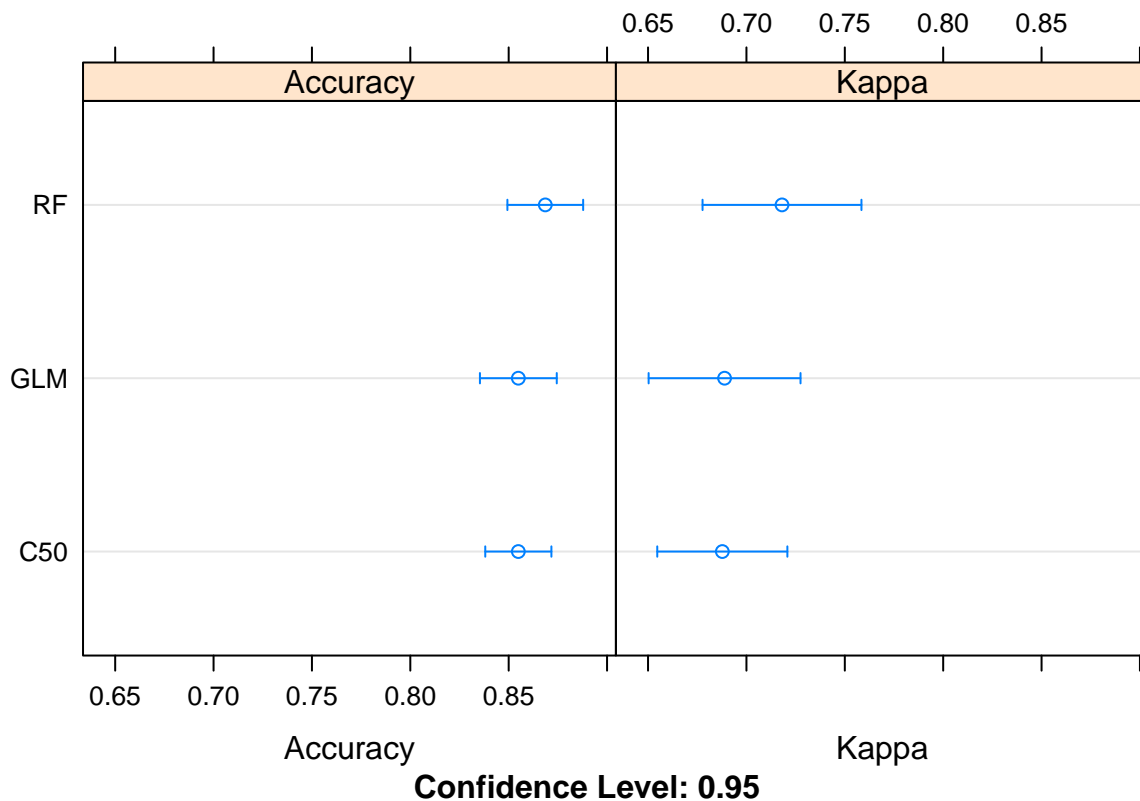
# Mostramos un resumen
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: C50, GLM, RF
## Number of resamples: 10
##
## Accuracy
##      Min.   1st Qu.   Median     Mean   3rd Qu.   Max. NA's
```

```
## C50 0.8091603 0.8435115 0.8620669 0.8549053 0.8678322 0.8846154 0
## GLM 0.8091603 0.8320611 0.8692308 0.8549053 0.8709663 0.8854962 0
## RF 0.8244275 0.8464474 0.8779138 0.8685997 0.8912214 0.9000000 0
##
## Kappa
##      Min.    1st Qu.    Median      Mean    3rd Qu.      Max. NA's
## C50 0.5974186 0.6674380 0.6947180 0.6877494 0.7144599 0.7513389 0
## GLM 0.5974186 0.6432831 0.7140890 0.6888915 0.7245203 0.7507294 0
## RF 0.6265956 0.6678607 0.7373031 0.7180824 0.7618813 0.7879548 0
# Y los boxplots y dotplots correspondientes
bwplot(results)
```



```
dotplot(results)
```



El modelo de random forest presenta una mayor precisión y un kappa más elevado, por lo que, de las tres técnicas revisadas, se considera esta como la más adecuada para este problema y será la que se utilice para construir el modelo final.

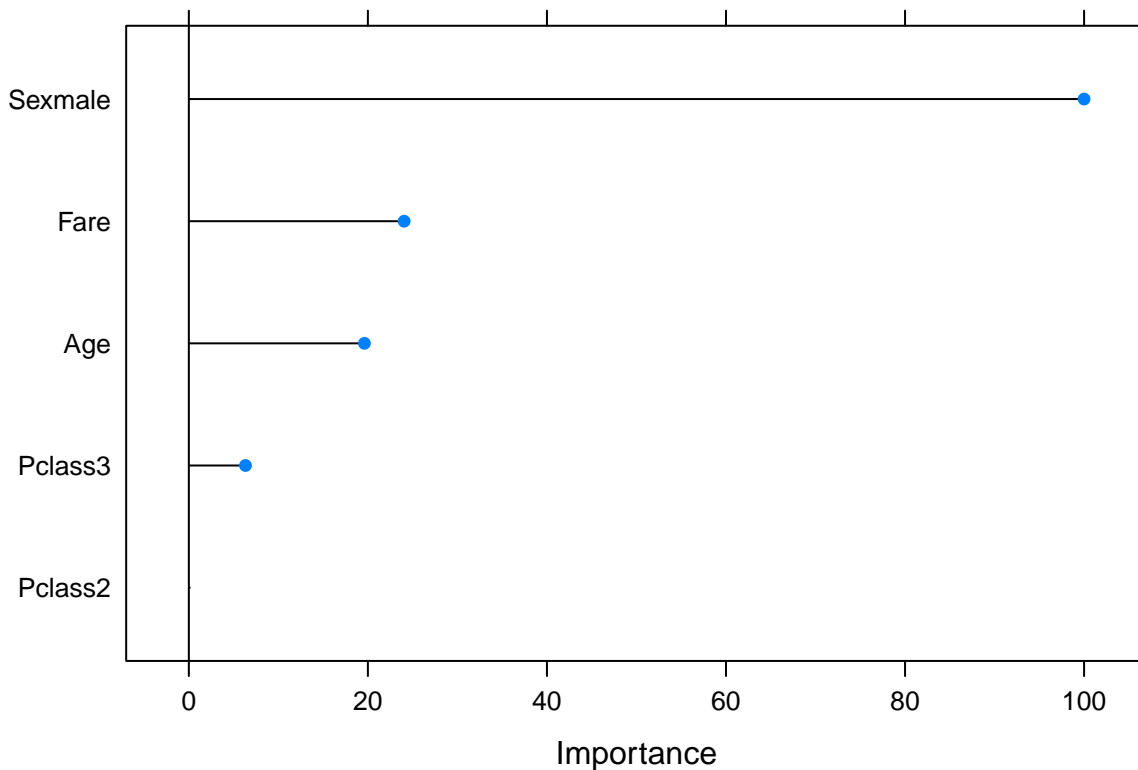
Una vez que se ha establecido que la técnica de random forest es la mejor de las evaluadas para resolver el problema planteado, debe construirse un nuevo modelo con todo el conjunto de datos disponible actualmente. Este nuevo modelo debería ser utilizado en el supuesto (totalmente teórico en el caso del Titanic) de que llegasen nuevas observaciones para las que fuese necesario predecir el valor de Survived.

```
## Estimamos la precisión con el modelo de regresión logística
set.seed(0806)
titanicFinalModel <- train(Survived ~ Pclass + Sex + Age + Fare,
                           data=titanicData, method="rf")
```

```
# Se muestran los resultados obtenidos
print(titanicFinalModel)
```

```
## Random Forest
##
## 1309 samples
## 4 predictor
## 2 classes: 'No', 'Si'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1309, 1309, 1309, 1309, 1309, 1309, ...
## Resampling results across tuning parameters:
```

```
##
## mtry Accuracy Kappa
## 2 0.8638290 0.7046928
## 3 0.8573066 0.6926645
## 5 0.8394034 0.6565147
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
plot(varImp(titanicFinalModel))
```



Conclusiones

Dado que aproximadamente el 63% de las personas que viajaban en el Titanic **no** sobrevivieron (valor mayoritario o moda, correspondiente a la variable Survived), cualquier modelo que se construya debe tener una precisión mayor que este porcentaje.

Tras la construcción y evaluación de varios modelos predictivos, hemos concluido que el mejor de ellos se basa en el algoritmo de random forest. Para evaluar los modelos se ha utilizado la técnica de validación cruzada 10-fold cross validation y se han contrastado los valores medios de la precisión y el índice kappa de cada uno de ellos.

En los tres modelos pre-seleccionados, se ha concluido que la variable más significativa es el sexo. De hecho, el modelo más preciso alcanzado con el árbol de decisión, utilizaba sólo esta variable. Esta sobresimplificación del problema, “las mujeres sobreviven, los hombres no”, conlleva a que el modelo sea descartado, aunque su precisión sea equiparable a la obtenida en la regresión logística, y a que se decida probar con otro tipo de técnicas.

Revisando las siguientes variables con mayor peso en la predicción, observamos algunas diferencias entre el modelo de regresión logística y el modelo de random forest:

```
## glm variable importance
##
##           Overall
## Sexmale 100.0000
## Pclass3  30.6383
## Age      0.6651
## Pclass2   0.0000

## rf variable importance
##
##           Overall
## Sexmale 100.000
## Fare    39.611
## Age     33.655
## Pclass3  6.325
## Pclass2  0.000
```

Mientras que para el modelo de regresión logística la siguiente variable en importancia es Pclass (clase del pasaje), para el de random forest las siguientes variables en importancia son Fare (coste del billete) y Age (edad del pasajero). De hecho, la variable Fare fue eliminada en el segundo modelo de regresión construido al concluir que no era una variable significativa.

A priori, parece natural que hubiera una relación directa entre las variables Fare y Pclass, ya que el precio del billete debería venir determinado por el tipo de clase. Sin embargo, en el apartado de análisis descriptivo visual ya se comprobó con varios gráficos que esta circunstancia no siempre es cierta, y por ello se mantuvieron ambas variables en el estudio.

Siguiendo con la revisión de la importancia de variables para random forest, se observa que el precio del billete tiene un peso superior al de la clase del pasaje. Esto parece sugerir que no es tanto la clase, sino los camarotes más caros lo que aumenta las posibilidades de supervivencia. Quizá la ubicación de estos camarotes fuese más cercana o tuviese mejores accesos a las cubiertas donde se encontraban los botes salvavidas, y de ahí su influencia en el modelo.

Considerando además la importancia y los resultados obtenidos para la edad y el sexo, se puede concluir que en el hundimiento del Titanic sí se siguió la norma no escrita de “las mujeres y los niños primero”.

Código fuente

El código generado para la realización de esta práctica se encuentra disponible en <https://github.com/azucenagm/Titanic>.

Bibliografía

Grolemund, Garrett; Wickham, Hadley (2016). *R for Data Science* (1ª ed.). Sebastopol: O'Reilly Media, Inc.

Teetor, Paul; Long, JD (2019). *R Cookbook* (2ª ed.). Sebastopol: O'Reilly Media, Inc.

Amat, Joaquín (2016, enero). “Análisis de la homogeneidad de varianza”. RPubS [artículo en línea]. [Fecha de consulta: 30 de mayo de 2019]. https://rpubs.com/Joaquin_AR/218466

Brownlee, Jason (2018, mayo). “A Gentle Introduction to k-fold Cross-Validation”. *Machine Learning Mastery* [Fecha de consulta: 31 de mayo de 2019]. <https://machinelearningmastery.com/k-fold-cross-validation/>

Brownlee, Jason (2014, septiembre). “How To Estimate Model Accuracy in R Using The Caret Package”. *Machine Learning Mastery* [Fecha de consulta: 31 de mayo de 2019]. <https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/>

Brownlee, Jason (2016, febrero). “Machine Learning Evaluation Metrics in R”. *Machine Learning Mastery* [Fecha de consulta: 1 de junio de 2019]. <https://machinelearningmastery.com/machine-learning-evaluation-metrics-in-r/>

Santana, Enmanuel (2014, noviembre). “Machine Learning con R: ejemplo de Random Forest”. *Apuntes R* [artículo en línea]. [Fecha de consulta: 22 de mayo de 2019]. <http://apuntes-r.blogspot.com/2014/11/ejemplo-de-random-forest.html>

“How do I interpret the AIC” (2018, abril). *R-bloggers* [artículo en línea]. [Fecha de consulta: 20 de mayo de 2019]. <https://www.r-bloggers.com/how-do-i-interpret-the-aic>

“Precios y descripción de los camarotes” (2012, diciembre). *El hundimiento* [artículo en línea]. [Fecha de consulta: 25 de mayo de 2019]. <https://elhundimiento.weebly.com/precios-clases-y-distribucion-de-los-camarotes.html>

“Qué incluía el billete más caro del Titanic” (2016, abril). *Traveler* [artículo en línea]. [Fecha de consulta: 25 de mayo de 2019]. <https://www.traveler.es/experiencias/articulos/billete-mas-caro-titanic/8724>

“RMS Titanic” (2019, mayo). *Wikipedia* [artículo en línea]. [Fecha de consulta: 25 de mayo de 2019]. https://es.wikipedia.org/wiki/RMS_Titanic%22

“Caret Package – A Practical Guide to Machine Learning in R” (2018, marzo). *Machine Learning Plus* [artículo en línea]. [Fecha de consulta: 1 de junio de 2019]. <https://www.machinelearningplus.com/machine-learning/caret-package/>

Contribuciones

Contribuciones	Firma
Investigación previa	AGM, JMD
Redacción de las respuestas	AGM, JMD
Desarrollo del código	AGM, JMD