PRA2 - Tipología y ciclo de vida de los datos

Jonathan Ayuquina y Martha Ayuquina

7 de junio, 2021

Contents

1 Descripción del dataset.

Desde Kaggle https://www.kaggle.com/c/titanic. obtenemos información sobre pasajeros del Titanic en dos partes, conjunto de entrenamiento y un conjunto de prueba:

- **test.csv** contiene 418 registros.
- **train.csv** contiene 891 registros.

Primero se cargan los ficheros *test.csv* y *train.csv* en los objetos correspondientes prueba y entrenamiento, respectivamente e identificando los valores NA.

```
# Cargamos el dataset https://www.kaggle.com/c/titanic/data?select=test.csv
titanic.test <- read.csv("test.csv", na.strings = c("NA", ""), stringsAsFactors = FALSE)
# Cargamos el dataset https://www.kaggle.com/c/titanic/data?select=train.csv
titanic.train <- read.csv("train.csv", na.strings = c("NA", ""), stringsAsFactors = FALSE)
```

A continuación, se muestra y examina las variables del objeto prueba, el resultado devuelve 418 observaciones y 11 variables: PassengerId, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked.

```
# Mostramos la estructura interna del objeto prueba
str(titanic.test)
```

```
## '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  NA NA NA NA ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Las principales estadísticas del objeto de prueba se presentan con la función summary().

```
# Producir resumen del resultado prueba
summary(titanic.test)
```

```
##   PassengerId         Pclass          Name               Sex
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##  NA's   :86
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:418         Length:418
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 35.627
##  3rd Qu.: 31.500
##  Max.   :512.329
##  NA's   :1
```

También se muestra y examina las variables del objeto entrenamiento, el resultado muestra 891 observaciones y 12 variables. Contiene los mismos atributos que el objeto de prueba más la variable adicional Survived.

```
# Mostramos la estructura interna del objeto entrenamiento
str(titanic.train)
```

```
## '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  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Las principales estadísticas del conjunto de entrenamiento se presentan con la función summary().

```
# Producir resumen del resultado entrenamiento
summary(titanic.train)
```

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

1.1 Importancia

Como se conoce, el barco RMS Titanic en abril de 1912 durante el viaje inaugural desde Southampton a Nueva York se hundió luego de chocar con un iceberg. Al no contar con suficientes botes salvavidas se produjo la muerte de 1502 de los 2224 pasajeros y la tripulación, es decir, hubo un 68% de personas fallecidas en el accidente.

Basado en lo expuesto, nos interesa efectuar un proyecto de análisis para aprendizaje automático y limpieza de datos por las características que tiene el conjunto de datos, esto es:

- Las observaciones se recogen de pasajeros del Titanic suficiente para realizar un modelo.
- Reúne variables cuantitativas y cualitativas que permite realizar análisis exploratorios.
- Tiene valores ausentes para ser tratados (eliminación o imputación) que influyen en el modelo.
- Requiere de proceso de limpieza y conversión de datos.

Luego de ello, podemos determinar los factores que influyeron en la supervivencia de los pasajeros y así, mediante modelos predictores y la evaluación de los mismos, estimar las probabilidades de supervivencia de un pasajero.

Este proyecto es relevante para estimar probabilidades de supervivencia de pasajeros ante accidentes marítimos, los cuales en la actualidad no están exentos de darse pese a que existen barcos con infraestructura segura. Entonces, es necesario establecer mecanismos de rescate y salvar vidas en situaciones de emergencia.

1.2 Pregunta/problema que se pretende responder

Es de suponer que ciertos grupos de personas tuvieron más probabilidades de supervivencia, siendo necesario un análisis para conocer los factores clave. Al respecto, surgen las siguientes preguntas:

¿Las mujeres tienen más probabilidades de supervivencia que los hombres?

¿Los familiares de los pasajeros tienen mayor probabilidad de supervivencia que los mismos pasajeros?

¿La clase de boleto que adquiere el pasajero influye para sobrevivir?

¿El punto de embarque afectó la probabilidad de supervivencia?

¿Es diferente la edad promedio de los sobrevivientes en comparación a la edad promedio de los fallecidos?

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

2.1 Integración

En este punto se realiza una exploración de los conjuntos para entender la información que tienen los datos que están divididos en dos: entrenamiento y prueba.

A breves rasgos se observa que ambos conjuntos tienen los mismos nombres de atributos y tipo de datos, excepto por la variable “Survived” que se encuentra en el conjunto de entrenamiento mas no en el conjunto de prueba.

La variable “Survived” marca la diferencia entre ambos conjuntos porque en el set de entrenamiento está antedicho si un pasajero es sobreviviente o no sobreviviente, mientras en el de pruebas se desconoce si un pasajero sobrevive o no.

Adicionalmente, el atributo “PassengerId” es un código único en ambos conjuntos que permitiría una integración vertical, sin embargo decidimos no combinarlo para analizarlos y limpiar por separado aunque se ingrese código de programación dos veces.

Como el atributo “Survived” consta solo en el conjunto de entrenamiento y siendo necesario para el resultado de nuestro análisis, le crearemos la variable “Survived” en el conjunto de prueba con valor vacío. La finalidad es emplear todas las variables cargadas y asignar los tipos que le corresponde almacenar según su naturaleza.

```
# Creamos la variable sobreviviente con el valor NA
titanic.test$Survived <- NA
```

Teniendo en cuenta que las variables están emparejadas, con el resumen previo se puede señalar que los conjuntos totalizan 1309 observaciones (891 registros para conjunto de entrenamiento y 418 registros para conjunto de prueba) y 12 variables que se detallan a continuación:

Variable	Descripción	Tipo
PassengerId	Identificador único del pasajero.	Cuantitativa discreta
Survived	Indicador si el pasajero sobrevivió “1” o no “0”.	Cualitativa nominal
Pclass	Clase de boleto del pasajero “1”, “2” o “3”.	Cualitativa ordinal
Name	Nombre del pasajero.	Cualitativa nominal
Sex	Sexo del pasajero con valores “male” y “female”.	Cualitativa nominal
Age	Edad del pasajero.	Cuantitativa discreta
SibSp	Número de hermanas/os, hermanastras/os, cónyuges en el barco.	Cuantitativa discreta
Parch	Número de padres e hijos que tenían a bordo los pasajeros.	Cuantitativa discreta
Ticket	Identificador del boleto.	Cualitativa nominal
Fare	Precio/tarifa pagado por el boleto.	Cuantitativa continua
Cabin	Identificación de la cabina/camarote asignado al pasajero.	Cualitativa nominal
Embarked	Puerto de embarque (Q=Queenstown, C=Cherburgo y S=Southampton)	Cualitativa nominal

2.2 Selección

Para dar respuesta a las interrogantes del numeral 1.2 y revisando de manera general el contenido de todas las variables puede comprenderse el efecto de la clase, sexo, edad, camarote, etc. en la supervivencia de los pasajeros.

La variable “Survived” es muy importante porque interesa estudiarla respecto a las demás para finalmente predecirla.

Notamos que las variables “PassengerId” y “Ticket” tienen muchos valores únicos y no serían relevantes para determinar la supervivencia de un pasajero, entonces se decide ignorar estos atributos.

Las variables “Age”, “Fare”, “SibSp” y “Parch” son una buena opción para limpieza de datos donde buscar valores extremos. En cambio, para “Age”, “Fare” tratar los valores nulos. No obstante, en la limpieza de datos con R podríamos detectar más atributos con caracterísitcas similares.

Más adelante, conforme analicemos cada atributo, determinaremos si una variable es necesaria en el análisis.

Contrario a la selección, se crearán los siguientes atributos:

- Designation: Variable que contiene la designación honorífica de personas, el cual se extrae del contenido del campo “Name” ya que a más del nombre del pasajero, el atributo contiene los tratamientos de cortesía luego de la coma y este detalle permite generar una nueva distribución de característica de las personas. Ej: Mr, Miss, Mrs, entre otros.
- FamilySize: Contiene el grupo de familia según su número. Para el efecto, empleamos la suma por registro de los atributos “SibSp” y “Parch” más 1 y catalogamos en: Solo si el resultado es 1; Familia pequeña si el resultado es mayor a 2 y menor a 5; Familia numerosa si es mayor a 4.
- AgeForGroup: Contiene la clasificación de las personas por edad. Considerando el atributo “Age”, si la edad es menor a 18 años se cataloga en el nuevo atributo como “Menor de edad”, caso contrario, si la edad es mayor a 18 se cataloga en el nuevo atributo como “Mayor de edad”.

3 Limpieza de los datos.

Con el nuevo conjunto de datos es más fácil limpiar los datos para el análisis y predicción.

3.1 ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Se verifica si hay valores perdidos dentro del dataset empleando la función is.na(), después accedemos el número de elementos vacíos en cada variable contando con colSums().

```
# Contar elementos vacíos de prueba
colSums(is.na(titanic.test))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          86           0
##       Parch      Ticket        Fare       Cabin    Embarked    Survived
##           0           0           1         327           0         418
```

```
# Contar elementos vacíos de entrenamiento
colSums(is.na(titanic.train))
```

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

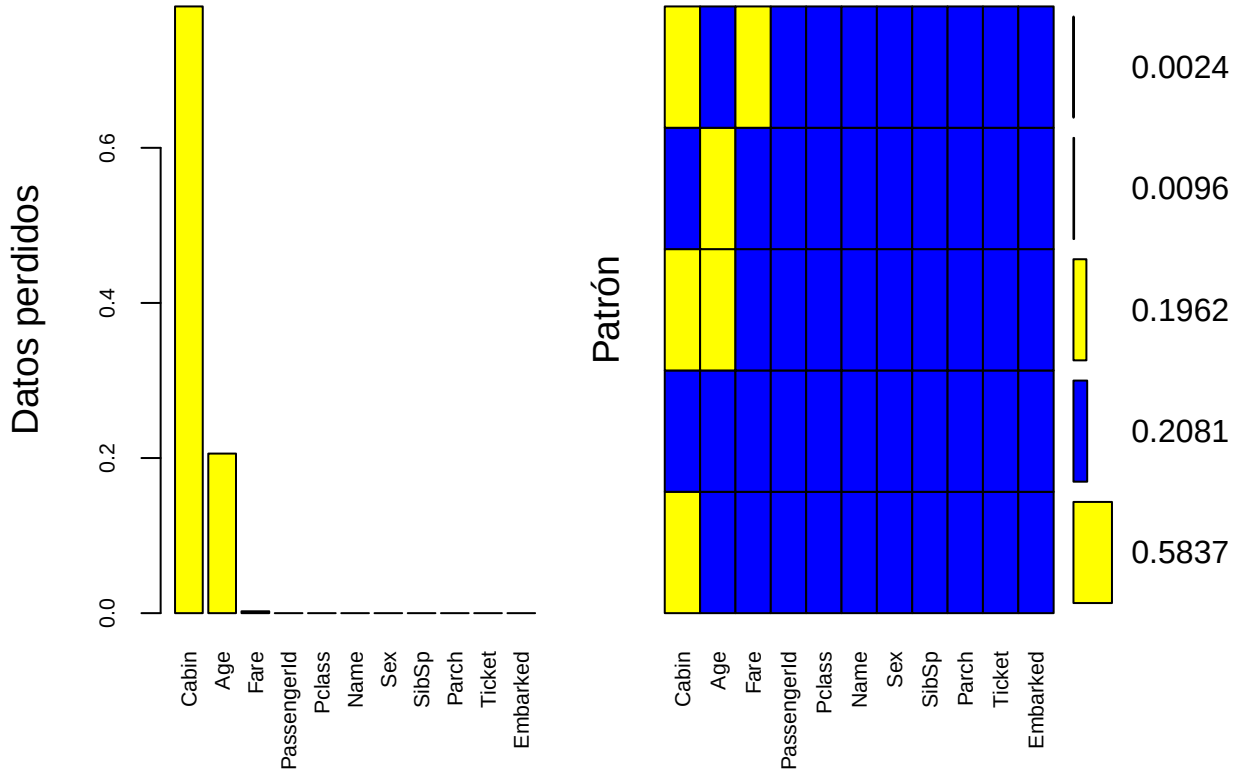
Lo anterior demuestra que sí hay elemento vacíos y para ello las gráficas de agregación son una herramienta útil para responder numéricamente la pregunta.

Representamos gráficamente los valores perdidos mediante la función aggr().

```
# Cargar librería VIM
library(VIM)
```

Se especifica el conjunto de prueba con que obtener la proporción de valores perdidos y combinaciones.

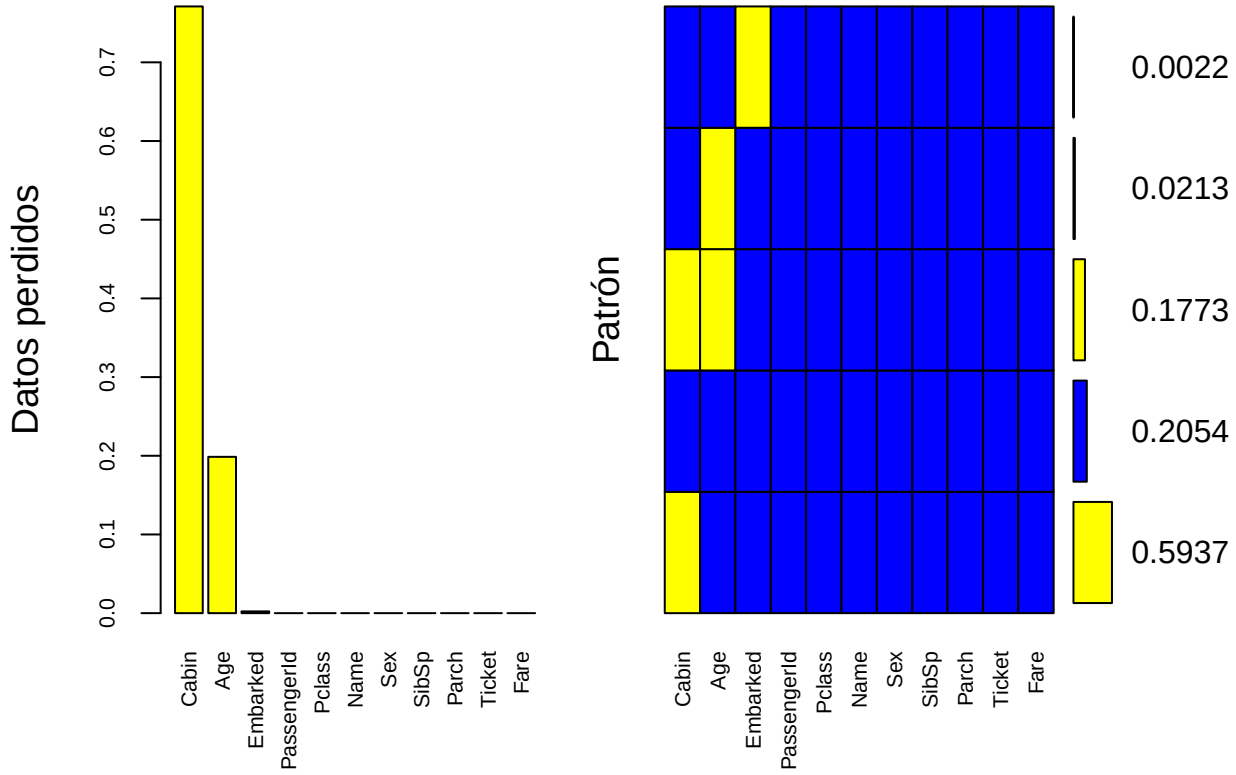
```
aggr(subset(titanic.test, select = -c(Survived)), col = c("blue", "yellow"),
 numbers = TRUE, sortVars = TRUE, cex.axis = 0.7, gap = 3,
 ylab = c("Datos perdidos", "Patrón"))
```




```
##
##  Variables sorted by number of missings:
##      Variable       Count
##         Cabin 0.782296651
##           Age 0.205741627
##          Fare 0.002392344
##   PassengerId 0.000000000
##        Pclass 0.000000000
##          Name 0.000000000
##           Sex 0.000000000
##         SibSp 0.000000000
##         Parch 0.000000000
##        Ticket 0.000000000
##      Embarked 0.000000000
```

La proporción de valores perdidos y combinaciones se obtiene del conjunto de entrenamiento.

```
aggr(subset(titanic.train, select = -c(Survived)), col = c("blue", "yellow"),
 numbers = TRUE, sortVars = TRUE, cex.axis = 0.7, gap = 3,
 ylab = c("Datos perdidos", "Patrón"))
```




```
##
##  Variables sorted by number of missings:
##     Variable       Count
##        Cabin 0.771043771
##          Age 0.198653199
##     Embarked 0.002244669
##  PassengerId 0.000000000
##       Pclass 0.000000000
##         Name 0.000000000
##          Sex 0.000000000
##        SibSp 0.000000000
##        Parch 0.000000000
##       Ticket 0.000000000
##         Fare 0.000000000
```

El resultado confirma los siguientes casos:

- Es claro que los valores perdidos aparecen en cuatro variables: “Cabin”, “Age”, “Fare” y “Embarked”.
- Las variables “Fare” y “Embarked” tienen pocos valores perdidos esto es 1 y 2 elementos y equivale al 0.2%.
- Hay una cantidad significativa de valores perdidos en las variables “Age” y “Cabin”, dado que tienen 263 y 1014 elementos ausentes, constituyendo alrededor del 20% y más del 60% respectivamente.
- La variable “Survived” denota 418 valores ausentes que es lo correcto para el conjunto de pruebas y se calcularán en el análisis concluyente.
- La combinación “Cabin” y “Age” tiene valores ausentes en ambos datasets.

A continuación se examina cada caso para aplicar el tratamiento según amerite.

Como la variable “Embarked” es cualitativa y tiene 2 valores perdidos, no preocupa reeemplazarlos con el valor de tendencia, es decir, el mayor valor de la tabla de distribución de los puertos “Q”, “C” y "S.

```
# Tabla de distribución de puertos
table(titanic.train$Embarked)
```

```
##
##   C   Q   S
## 168  77 644
```

La mayor cantidad de pasajeros proviene del puerto Southampton “S” con 644 personas y se considerará esta tendencia para reemplazar la cadena faltante con “S”.

```
# Reemplazar NA con puerto Southampton "S" para mantener la tendencia
titanic.train$Embarked[is.na(titanic.train$Embarked)] <- "S"
```

Así mismo la variable cuantitativa “Fare” tiene 1 valor perdido y puede ser reemplazado por la mediana de valores de “Fare”, esto es $14.45.

```
# Reemplazar los registros perdidos por la mediana
fare.median <- median(titanic.test$Fare, na.rm = TRUE)
fare.median
```

```
## [1] 14.4542
```

```
titanic.test$Fare[is.na(titanic.test$Fare)] <- fare.median
```

Examinando la variable cualitativa “Cabin” tiene muchos valores diferentes y 1014 (327+687) están perdidos, no se tiene cómo asignarle valor, entonces se decide mantener NA e ignorar este atributo completamente.

```
# Leer las últimas filas del dataset
tail(titanic.test$Cabin)
```

```
## [1] NA     NA     "C105" NA     NA     NA
```

```
tail(titanic.train$Cabin)
```

```
## [1] NA     NA     "B42"  NA     "C148" NA
```

La variable cuantitativa “Age” podría tener la misma gestión que “Fare” para llenarse con la mediana, pero tiene 263 (86+177) valores NA y emplear el método no podría ser preciso. Entonces, utilizaremos un modelo de aproximación en base a las variables existentes para predecir los elementos vacíos.

Previamente se revisan las variables para constatar si hay alguna que contribuya al análisis de predicción de la edad faltante y transformarla.

Por mencionar, en la variable “Name” se encuentran los nombres de pasajeros que podemos explorar con la función head().

```
# Devolver primeras filas del dataset
head(titanic.test$Name)
```

```
## [1] "Kelly, Mr. James"
## [2] "Wilkes, Mrs. James (Ellen Needs)"
## [3] "Myles, Mr. Thomas Francis"
## [4] "Wirz, Mr. Albert"
## [5] "Hirvonen, Mrs. Alexander (Helga E Lindqvist)"
## [6] "Svensson, Mr. Johan Cervin"
```

```
head(titanic.train$Name)
```

```
## [1] "Braund, Mr. Owen Harris"
```

```
## [2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## [3] "Heikkinen, Miss. Laina"
## [4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
## [5] "Allen, Mr. William Henry"
## [6] "Moran, Mr. James"
```

```
# Contar los nombres de pasajeros
length(unique(titanic.test$Name))
```

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

```
length(unique(titanic.train$Name))
```

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

El resultado muestra el nombre de 1308 (418+891) personas y es de resaltar que el nombre incluye la designación honorífica del pasajero cuando se registró, como puede ser: "Mr.", "Mrs.", "Miss", etc. Con esto se puede discretizar en otra variable esta clase para observar la frecuencia absoluta de las Variables "Sex" y "Designation".

```
# Crear variable con la designación tomada del nombre de pasajeros de prueba
titanic.test$Designation <- gsub("^.*, (.*?)\\..*$", "\\1", titanic.test$Name)

# Tabla de frecuencia absoluta
table(titanic.test$Sex, titanic.test$Designation)
```

```
##
##          Col Dona  Dr Master Miss  Mr Mrs  Ms Rev
##   female   0    1   0      0   78   0  72   1   0
##   male     2    0   1     21    0 240   0   0   2
```

```
# Crear variable con la designación tomada del nombre de pasajeros de entrenamiento
titanic.train$Designation <- gsub("^.*, (.*?)\\..*$", "\\1", titanic.train$Name)

# Tabla de frecuencia absoluta
table(titanic.train$Sex, titanic.train$Designation)
```

```
##
##          Capt Col Don  Dr Jonkheer Lady Major Master Miss Mlle Mme  Mr Mrs  Ms
##   female    0   0   0   1        0    1     0      0  182    2   1   0 125   1
##   male      1   2   1   6        1    0     2     40    0    0   0 517   0   0
##
##          Rev Sir the Countess
##   female   0   0            1
##   male     6   1            0
```

La tabla muestra valores en las clases "Miss", "Mrs", "Mr." y en otras que coinciden con estas designaciones. Siendo así, se podría reducir las etiquetas, por ejemplo: la clase "Mlle" y "Ms" son lo mismo que "Miss". La clase "Mme" es lo mismo que "Mrs". Los grupos con menor cantidad de repetición podrían reunirse en la categoría "Other".

```
# Reducir la designación de pasajeros de prueba
titanic.test$Designation[titanic.test$Designation %in% c("Mlle", "Ms")] <- "Miss"
titanic.test$Designation[titanic.test$Designation %in% c("Mme")] <- "Mrs"
titanic.test$Designation[titanic.test$Designation %in% c("Capt", "Col", "Don",
 "Dona", "Dr", "Jonkheer", "Lady", "Major", "Rev", "Sir", "the Countess")] <-
 "Other"
```

```
# Reducir la designación de pasajeros de entrenamiento
titanic.train$Designation[titanic.train$Designation %in% c("Mlle", "Ms")] <- "Miss"
titanic.train$Designation[titanic.train$Designation %in% c("Mme")] <- "Mrs"
titanic.train$Designation[titanic.train$Designation %in% c("Capt", "Col", "Don",
 "Dona", "Dr", "Jonkheer", "Lady", "Major", "Rev", "Sir", "the Countess")] <-
 "Other"
```

Se muestra el resultado de combinar la designación en una tabla que contenga las agrupaciones efectuadas.

```
# Tabla de frecuencia sexo y título
table(titanic.test$Sex, titanic.test$Designation)
```

```
##
##          Master Miss  Mr Mrs Other
##   female      0   79   0  72     1
##   male       21    0 240   0     5
```

```
# Tabla de frecuencia sexo y título
table(titanic.train$Sex, titanic.train$Designation)
```

```
##
##          Master Miss  Mr Mrs Other
##   female      0  185   0 126     3
##   male       40    0 517   0    20
```

Por otro lado, la cantidad de las variables "SibSp" y "Parch", corresponden a familiares de personas a bordo del barco y pueden sumarse adicionalmente a 1 pasajero para formar otra que simplifique el conjunto de familia.

```
# Crear variable tamaño de familiares pasajeros de prueba
titanic.test$FamilySize <- titanic.test$SibSp+titanic.test$Parch+1
```

```
# Crear variable tamaño de familiares pasajeros entrenamiento
titanic.train$FamilySize <- titanic.train$SibSp+titanic.train$Parch+1
```

Según esto, con el número de los integrantes de la familia discretizamos el tamaño de familia en: "Alone" si solo viaja el pasajero (sin familia); "Small" si los familiares y el pasajero conforman un grupo máximo de 4 personas y "Large" si el número de familiares es mayor a 4.

```
# Crear variable con el concepto de tamaño de familia
titanic.test$FamilyGroup[titanic.test$FamilySize==1] <- "Alone"
titanic.test$FamilyGroup[titanic.test$FamilySize>1 & titanic.test$FamilySize<=4] <- "Small"
titanic.test$FamilyGroup[titanic.test$FamilySize>4] <- "Large"
```

```
# Crear variable con el concepto de tamaño de familia
titanic.train$FamilyGroup[titanic.train$FamilySize==1] <- "Alone"
titanic.train$FamilyGroup[titanic.train$FamilySize>1 & titanic.train$FamilySize<=4] <- "Small"
titanic.train$FamilyGroup[titanic.train$FamilySize>4] <- "Large"
```

Retomando el relleno de valores perdidos de la variable "Age" y dado que obtuvimos otras variables, se aplica la predicción o imputación usando un método no paramétrico. La fórmula es introducida en la función missForest() para obtener un modelo ajustado a lo siguiente.

```
library(missForest)
```

```
age.imputation.test <- titanic.test[, c("Pclass", "Sex", "Age", "Fare", "Embarked",
 "Designation",  "FamilyGroup")]
age.imputation.train <- titanic.train[, c("Pclass", "Sex", "Age", "Fare", "Embarked",
 "Designation",  "FamilyGroup")]
```

```r
# Discretizar las variables con pocas clases
cols <- c("Sex", "Embarked", "Designation",  "FamilyGroup")
for(i in cols) {
  age.imputation.test[,i] <- as.factor(age.imputation.test[,i])
  age.imputation.train[,i] <- as.factor(age.imputation.train[,i])
}
```

```r
# Mostrar el resumen de valores antes de imputación
summary(age.imputation.test$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.17   21.00   27.00   30.27   39.00   76.00      86
```

```r
summary(age.imputation.train$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

```r
# Imputar valores perdidos de prueba usando todos los parámetros
set.seed(123)
age.imp.test <- missForest(age.imputation.test, variablewise = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
```

```r
# Imputar valores perdidos de entrenamiento usando todos los parámetros
set.seed(123)
age.imp.train <- missForest(age.imputation.train, variablewise = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
```

```r
# Mostrar el resumen de valores imputados
summary(age.imp.test$ximp$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.17   22.00   26.96   29.70   36.38   76.00
```

```r
summary(age.imp.train$ximp$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   21.03   28.97   29.68   36.71   80.00
```

```r
# Verificar error de imputación
age.imp.test$OOBerror
```

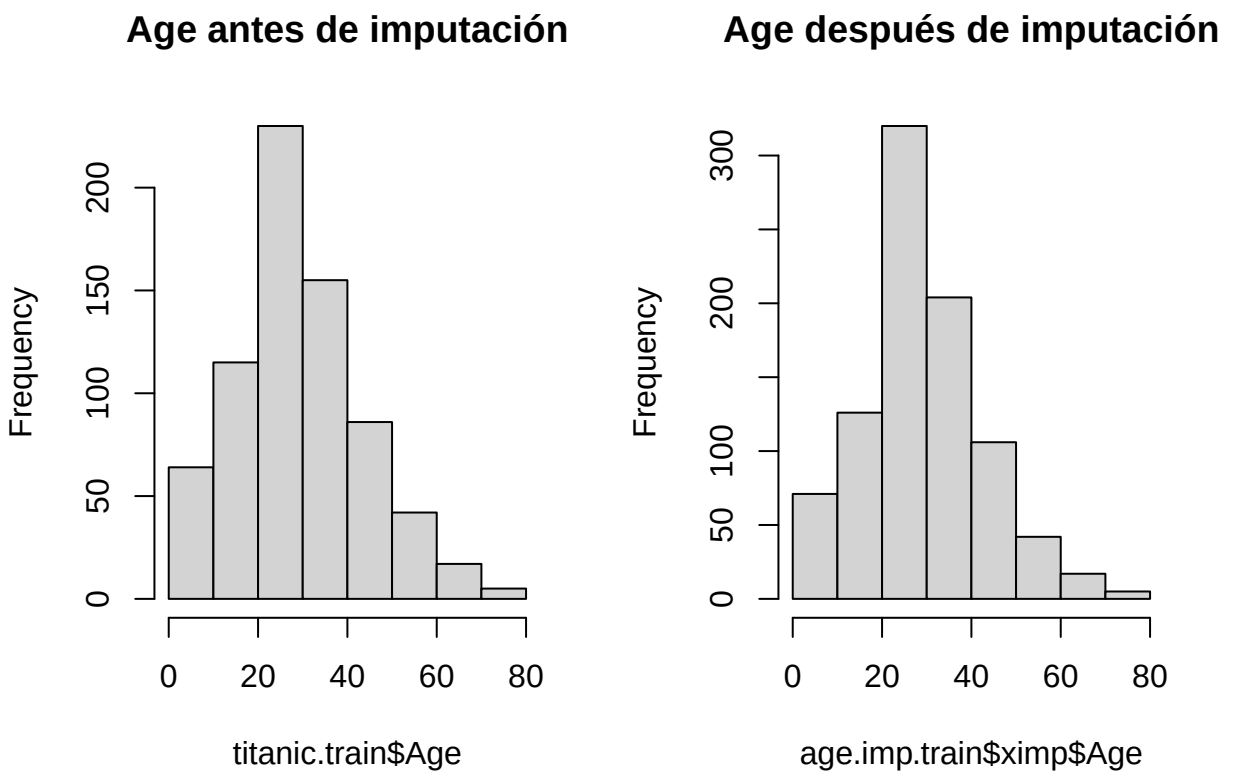
```
##      MSE      PFC      MSE      MSE      PFC      PFC      PFC
##   0.0000   0.0000 119.4307   0.0000   0.0000   0.0000   0.0000
```

```r
age.imp.train$OOBerror
```

```
##      MSE      PFC      MSE      MSE      PFC      PFC      PFC
##   0.0000   0.0000 125.9857   0.0000   0.0000   0.0000   0.0000
```

Se comprueba el resultado de la imputación adicionalmente del resumen estadístico en uno de los conjuntos.

```r
# Graficar resultado de imputación para segmento de entrenamiento
par(mfrow = c(1, 2))
hist(titanic.train$Age, main = "Age antes de imputación")
hist(age.imp.train$ximp$Age, main = "Age después de imputación")
```



La calidad de la imputación es excelente considerando el patrón de distribución es idéntico antes y después.

```r
titanic.test$Age <- age.imp.test$ximp$Age
titanic.train$Age <- age.imp.train$ximp$Age
```

Podemos aprovechar la variable "Age" completa para dividirla en dos grupos a los pasajeros: los de menos de 18 años como niños y los mayores como adultos.

```r
# Segmentar pasajeros menores y mayores de edad de prueba
titanic.test$AgeForGroup[titanic.test$Age<18] <- "Child"
titanic.test$AgeForGroup[titanic.test$Age>=18] <- "Adult"
```

```r
# Segmentar pasajeros menores y mayores de edad de entrenamiento
titanic.train$AgeForGroup[titanic.train$Age<18] <- "Child"
titanic.train$AgeForGroup[titanic.train$Age>=18] <- "Adult"
```

3.2 Identificación y tratamiento de valores extremos.

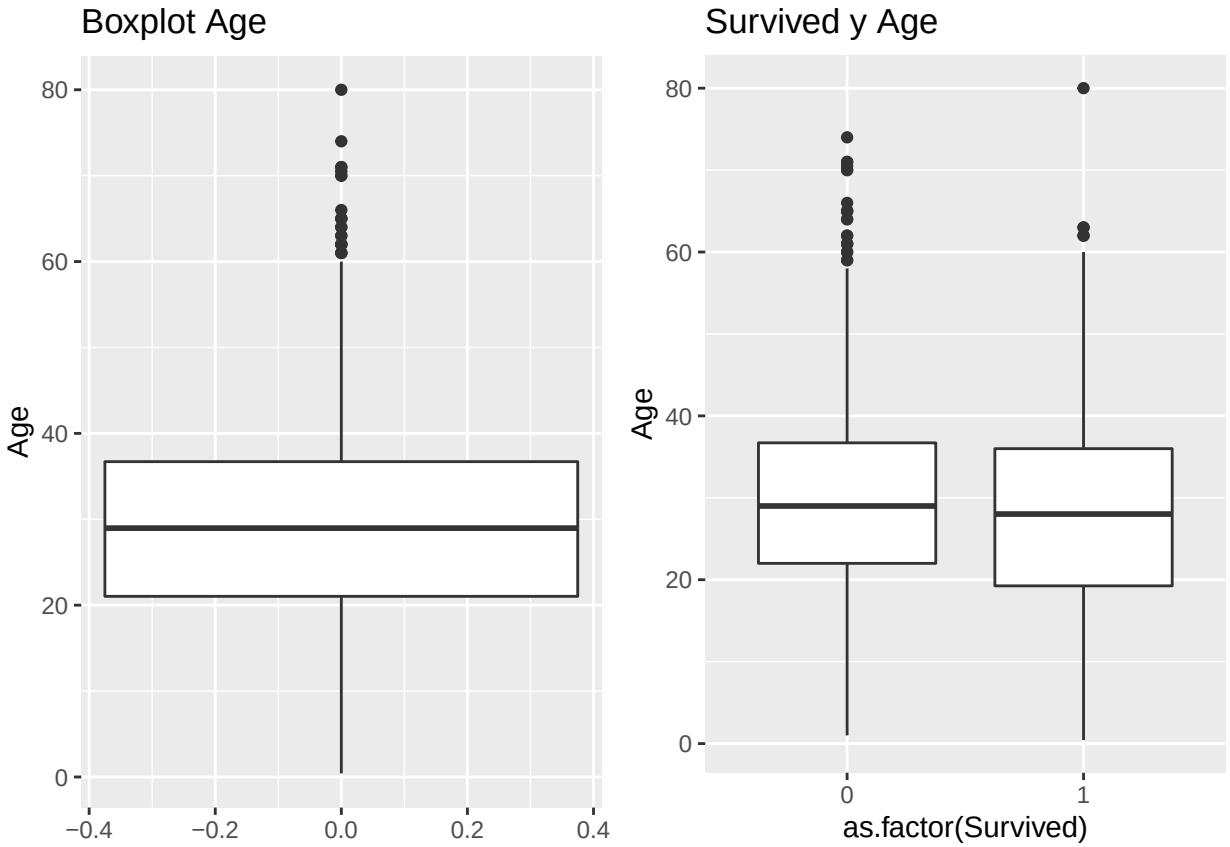
La identificación de valores extremos se realiza a través de diagramas de cajas para interpretar los defectos en los datos de entrenamiento de las variables cuantitativas "Age" y "Fare".

Inicialmente se cargan las librerías de R para realizar los gráficos entre variables.

```r
# Cargar librerías ggplot2, grid y gridExtra
library(ggplot2)
library(grid)
library(gridExtra)
```

Se muestra los diagramas de la variable "Age" para ver cuántos valores están fuera de la caja, a través de la función ggplot(). Asimismo se analiza con la variable resultado "Survived".

```r
age.plot <- ggplot(titanic.train, aes(y=Age))+
  geom_boxplot()+ggtitle("Boxplot Age")

age.survived.plot <- ggplot(titanic.train, aes(x= as.factor(Survived), y=Age))+
  geom_boxplot()+ggtitle("Survived y Age")

grid.arrange(age.plot, age.survived.plot, ncol = 2)
```



En el diagrama se percibe los siguientes aspectos:

- La distribución oscila de manera asimétrica en rango de 0 a 60 años aproximadamente.
- Las edades de los grupos entre sobrevivientes y fallecidos se distribuye en forma similar.
- Un 75% de los pasajeros que compone la muestra de edades es menor de 40 años.
- Hay algunas edades atípicas distantes del límite superior, mayores a 60.
- El grupo sobreviviente tiene un sesgo en los pasajeros menores de 30, al contrario del grupo fallecido que se concentra en los de mayor edad, quizá porque usaron un protocolo de mujeres y niños para salvarlos primero.

La función boxplot.stats() permite obtener las puntos relevantes de la caja, estos son la edad mínima y máxima.

```
# Obtener los valores del diagrama
boxplot.stats(titanic.train$Age)$stats
```

```
## [1]  0.42000 21.03005 28.97018 36.70917 60.00000
```

Con la función boxplot.stats() se detalla los valores de los que se trata.

```
# Mostrar edades fuera del extremo
boxplot.stats(titanic.train$Age)$out
```

```
##  [1] 66.0 65.0 71.0 70.5 61.0 62.0 63.0 65.0 61.0 64.0 65.0 63.0 71.0 64.0 62.0
## [16] 62.0 61.0 80.0 70.0 70.0 62.0 74.0
```

```
length(boxplot.stats(titanic.train$Age)$out)
```

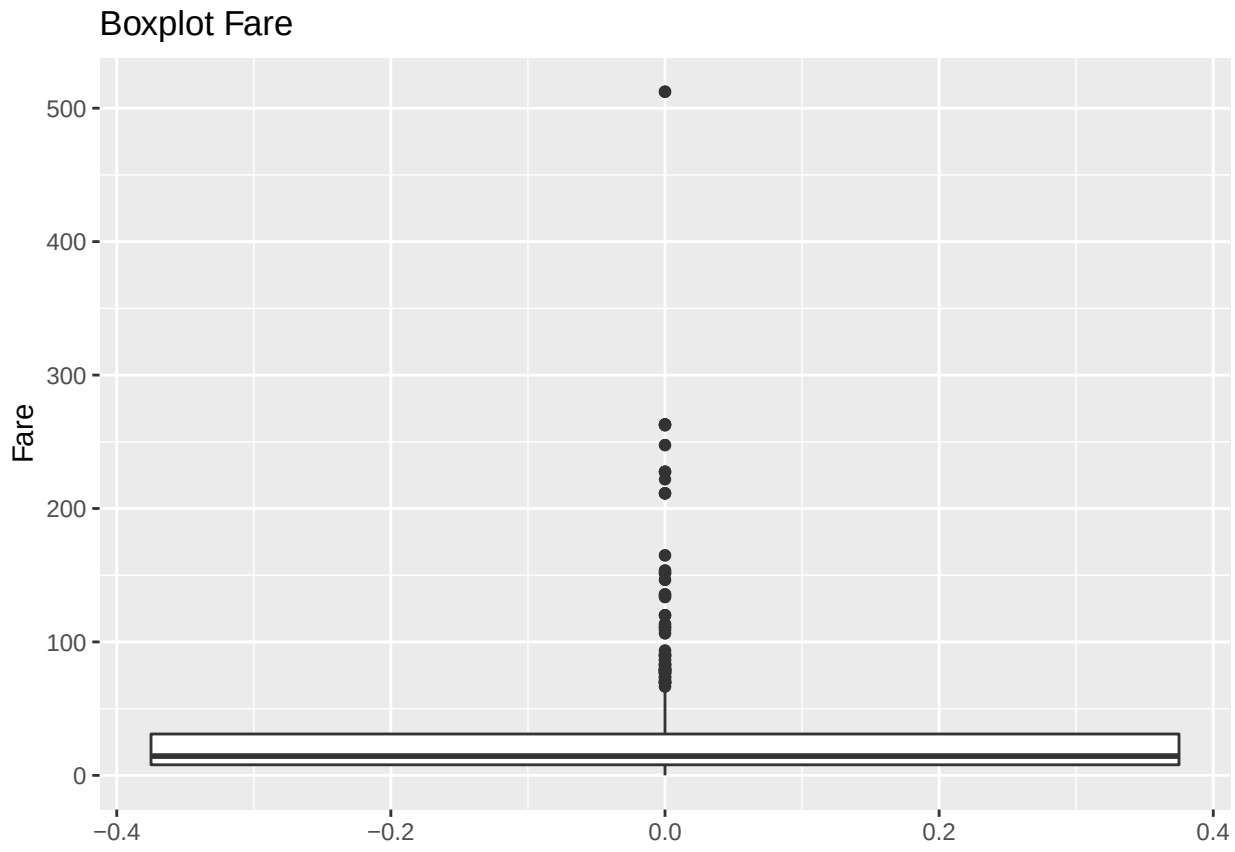
```
## [1] 22
```

La salida ubica 22 personas del grupo etario vejez fuera de la distribución, es decir, del extremo superior del bigote (65). Esto no significa que sean incoherentes y no representa una condición suficiente para reemplazarlos.

Las edades imputadas son decimales y parecen naturales si no se toma en cuenta los decimales añadidos, por lo que resolvemos mantenerlos.

Analizamos también los valores que estén fuera para la variable "Fare" con un diagrama de cajas.

```
ggplot(titanic.train, aes(y = Fare), na.rm = TRUE)+
  geom_boxplot()+ggtitle("Boxplot Fare")
```



Evidenciamos que existen 171 valores extremos considerando el valor extremo superior del bigote (65).

```
# Obtener el extremo superior del diagrama
upper.fare <- boxplot.stats(titanic.train$Fare)$stats[5]
upper.fare
```

```
## [1] 65
```

```
# Filtrar valores menores que $65
length(titanic.train$Fare[titanic.train$Fare>upper.fare])
```

```
## [1] 116
```

Evaluando el valor mínimo y máximo de tarifa, se obtiene 0 y 512.33 respectivamente. Las tarifas con valor cero suponen que los pasajeros no pagaron por su ticket de entrada porque fueran políticos o personas con importancia económica o también personas que recibieron boletos gratuitos.

```
#Obtengo valor mínimo y máximo
min(titanic.train$Fare)
```

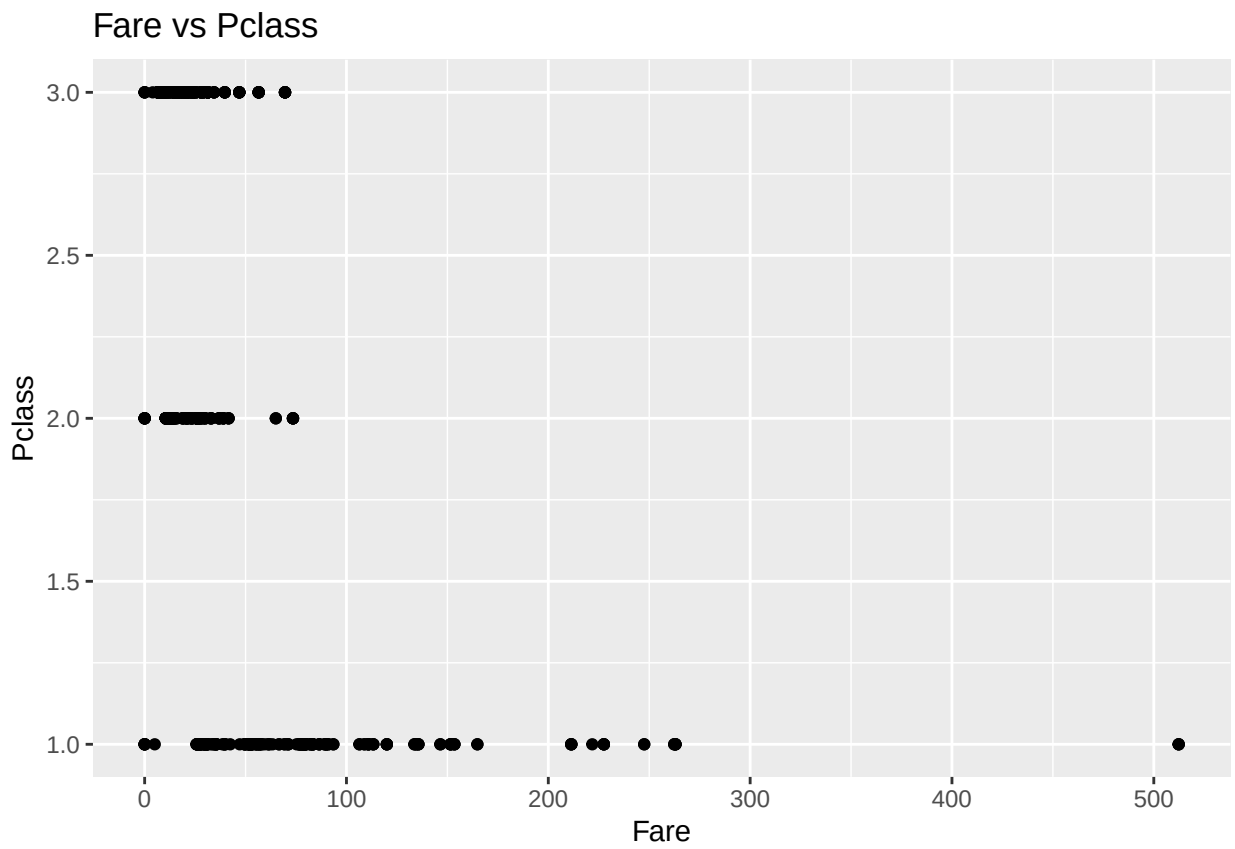
```
## [1] 0
```

```
max(titanic.train$Fare)
```

```
## [1] 512.3292
```

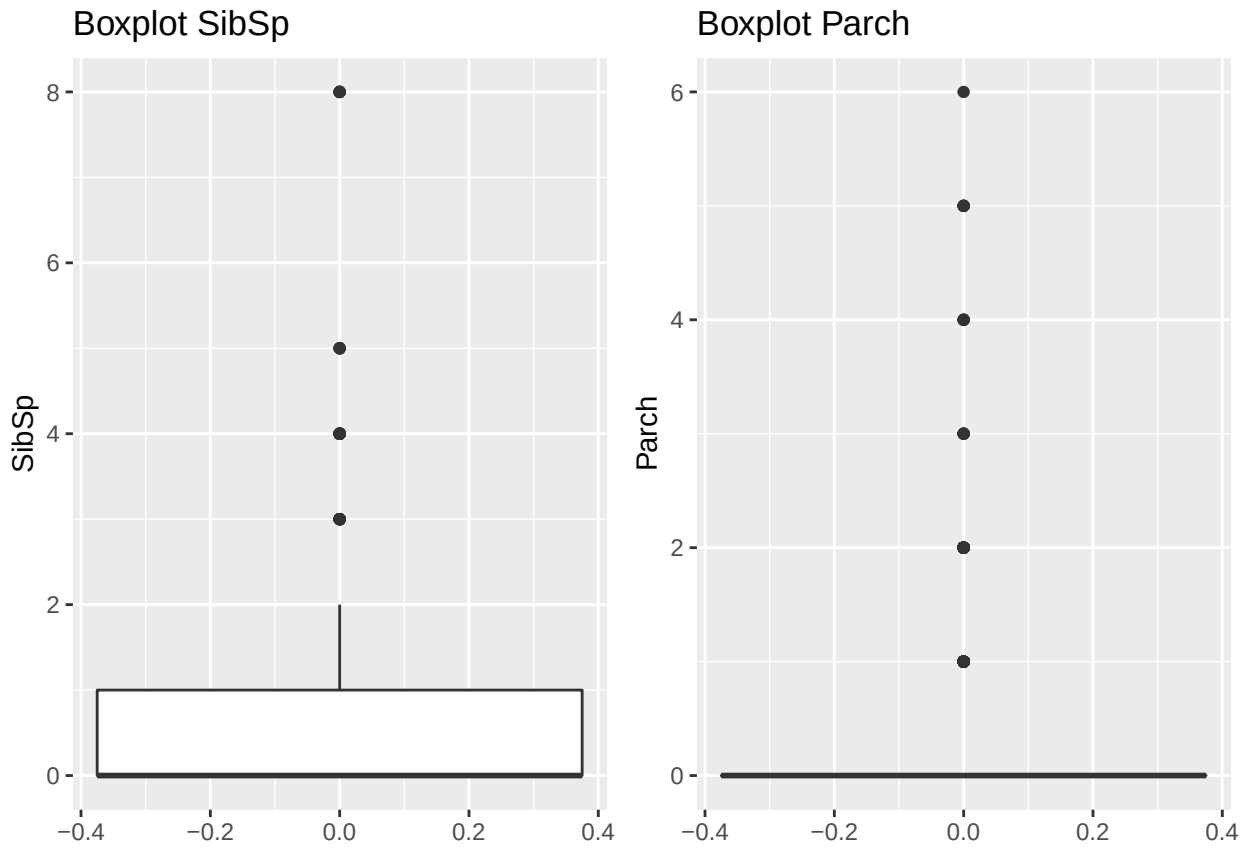
Se realiza una gráfica de comparación entre la tarifa y la clase de boleto para descartar el registro de tarifa cero y se constata que, para las tres clases existen tarifa cero; sin embargo, es posible notar una relación entre ambas variables determinando que no es necesario modificar la tarifa porque son datos legítimos. Entonces, consideramos los outliers de la variable Fare en nuestro análisis.

```
ggplot(titanic.train, aes(x = Fare, y = Pclass))+
  geom_point()+ggtitle("Fare vs Pclass")
```



Las variables "SibSp" y "Parch" demuestran número de familiares fuera de rango, sin embargo no hay cómo contrastarlo y se justifica cuando el registro señala existencia de familias enteras a bordo.

```
# Mostrar número de familiares fuera del extremo
sibSp.plot <- ggplot(titanic.train, aes(y = SibSp))+
  geom_boxplot()+ggtitle("Boxplot SibSp")

parch.plot <- ggplot(titanic.train, aes(y = Parch))+
  geom_boxplot()+ggtitle("Boxplot Parch")

grid.arrange(sibSp.plot, parch.plot, ncol = 2)
```



Se convierte las variables a los tipos correctos. La variable "Survived" no conviene almacenarla en formato numérico ya que esto puede llevar a errores como tratar de calcular la media, por lo que se convierte a factor.

```
cols <- c("Pclass", "Sex", "Embarked", "Designation", "FamilySize", "FamilyGroup",
 "AgeForGroup", "Survived")
for(i in cols) {
  titanic.test[,i] <- as.factor(titanic.test[,i])
  titanic.train[,i] <- as.factor(titanic.train[,i])
}
```

Finalmente del conjunto original especificamos nuevos objetos con la configuración realizada a las variables.

```
new.titanic.test <- titanic.test
new.titanic.train <- titanic.train
```

4 Análisis de los datos.

4.1 Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Como nuestro interés es la respuesta a las preguntas planteadas en el numeral 1 que se enfoca en la explicación de la supervivencia de los pasajeros (variable respuesta), determinamos la importancia de los predictores.

Para seleccionar los grupos de datos a comparar consideraremos los siguientes pasos:

1. Análisis estadístico descriptivo para tener un resumen de cada uno de los atributos del conjunto. Aunque es de aclarar que, durante la limpieza de datos ya realizamos una visión general para estudiarlos.
2. Análisis estadístico inferencial, mediante el cual disponiendo de una muestra de datos, procederemos a:

- Conocer que los datos siguen una distribución normal y homocedasticidad.
- Efectuar el análisis de correlación entre pares de variables.

3. Crear tres modelos predictivos.

- Evaluar la exactitud de los modelos.
- Comparar los modelos para seleccionar el mejor modelo predictivo.

4.2 Comprobación de la normalidad y homogeneidad de la varianza.

4.2.1 Comprobación de la normalidad

Para estudiar si la muestra proviene de una población con distribución normal se disponen de tres herramientas: histogramas, gráfico de cuantil cuantil y prueba de hipótesis.

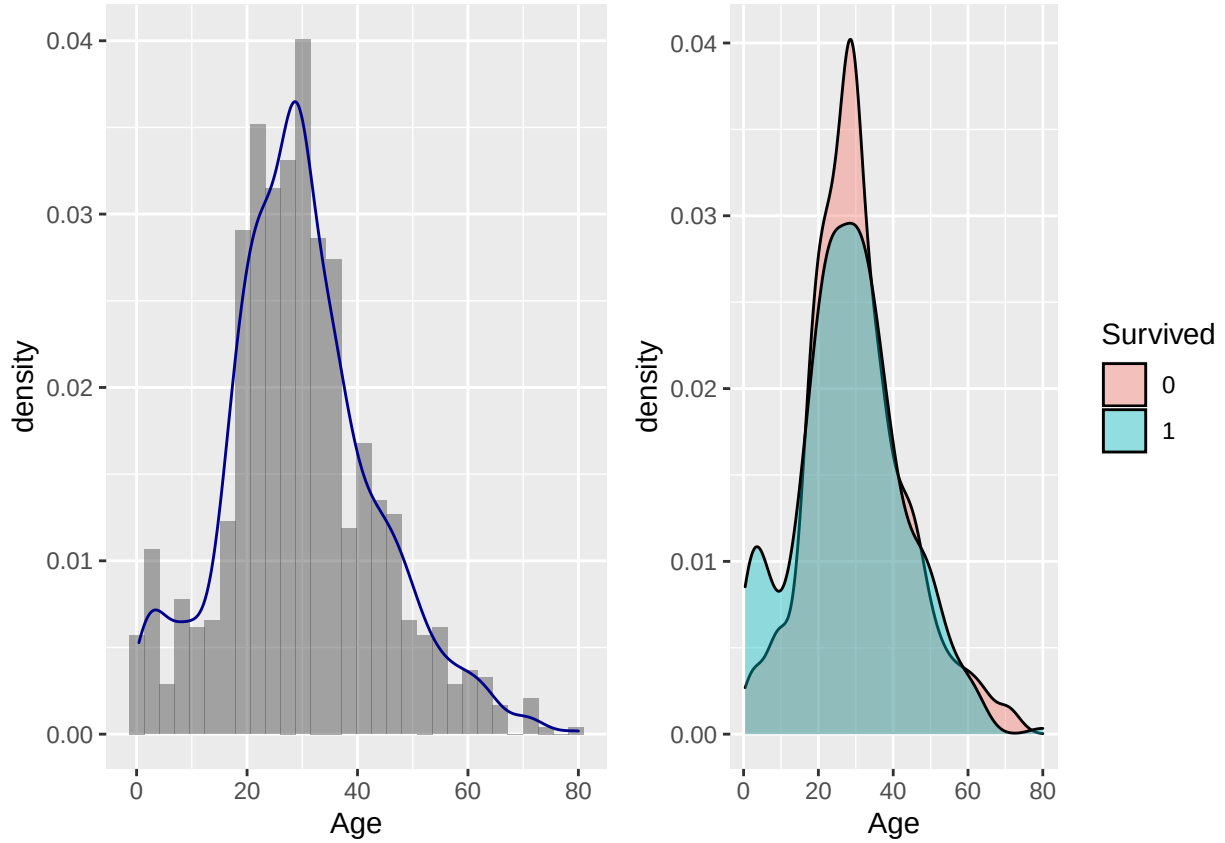
En el histograma o gráfico de densidad se explora la normalidad presente mediante un patrón más o menos simétrico.

A continuación se construye el histograma y gráfico de densidad de los datos considerando que de los pasajeros se desea saber si la variable "Age" tiene una distribución normal.

Además se hace la relación con la variable "Survived" para ver la condición de supervivencia.

```
age.density.plot <- ggplot(new.titanic.train, aes(x = Age))+
  geom_histogram(aes(y = ..density..), alpha = 0.5, position = "identity")+
  geom_density(color = "darkblue", alpha = 0.2)

age.survived.density.plot <- ggplot(new.titanic.train, aes(x = Age, fill = Survived))+
  geom_density(alpha = 0.4)

grid.arrange(age.density.plot, age.survived.density.plot, ncol = 2)
```

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



En el gráfico se observa:

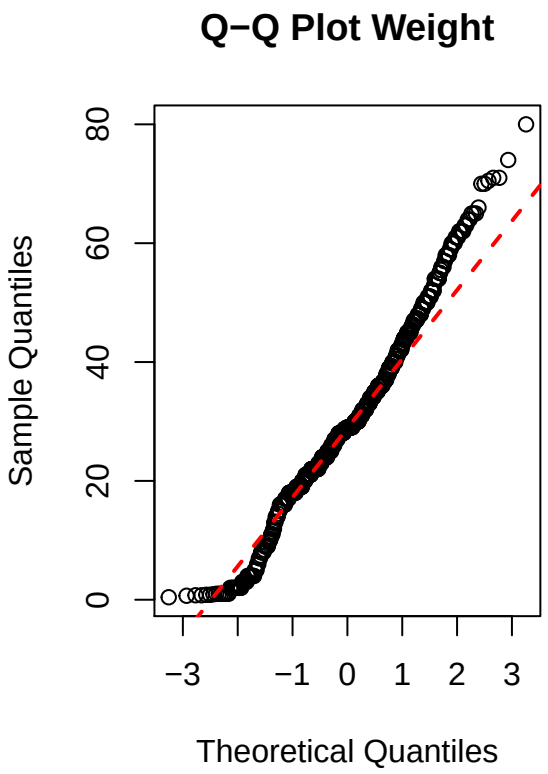
- Que las densidades no son perfectamente simétricas.
- La mayor densidad se encuentra en el intervalo de 20 a 30 años.
- Hay un sesgo hacia los pasajeros de menor edad, esto hace que se desconfíe de la normalidad de los datos.
- La distribución de edad de los pasajeros es similar entre el grupo de sobrevivientes y fallecidos.

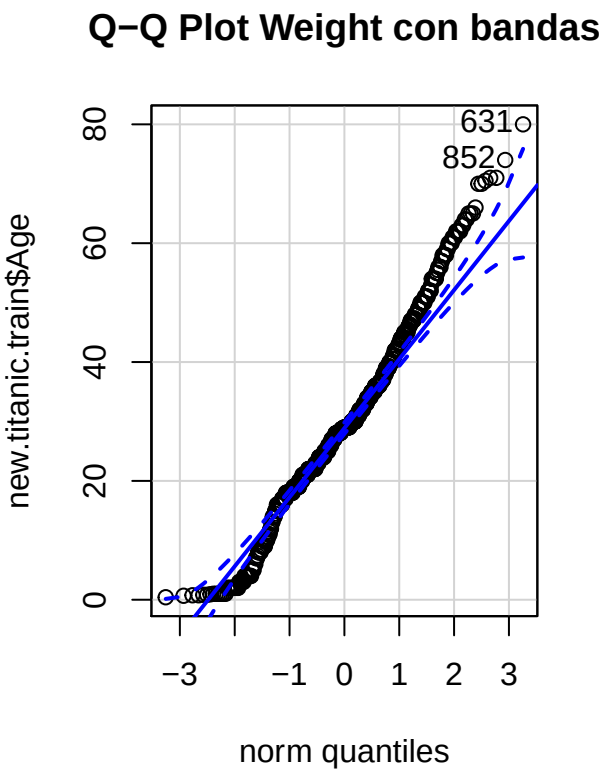
Otra herramienta como el QQplot podría aportar una mejor conclusión de la normalidad si los datos están perfectamente alineados a la línea de referencia, para ello se carga las librerías necesarias.

```
# Cargar librería car
library(car)
```

La función qqnorm() sirve para construir el QQplot y la función qqline() agrega una línea de referencia que ayuda a interpretar el gráfico de cierta distribución. La función qqplot() del paquete car permite mostrar bandas para los puntos del gráfico.

```
par(mfrow = c(1,2))
qqnorm(new.titanic.train$Age, main = "Q-Q Plot Weight")
qqline(new.titanic.train$Age, col="red", lwd = 2, lty = 2)
qqPlot(new.titanic.train$Age, main = "Q-Q Plot Weight con bandas")
```






```
## [1] 631 852
```

La figura del QQplot explica:

- Los puntos de edad de pasajeros están desalineados.
- Mientras que con las bandas no todos los puntos están dentro.
- Esto lleva a rechazar la hipótesis de normalidad.

Por medio de las pruebas de normalidad se explora la normalidad del conjunto de datos formulando las hipótesis:

Hipótesis nula (H_0): La distribución es normal

Hipótesis alternativa (H_1): La distribución NO es normal

De los tipos de pruebas existentes se aplica la prueba de normalidad Shapiro Wilks en R con nivel de significancia del 5%.

```
shapiro.test(new.titanic.train$Age)
```

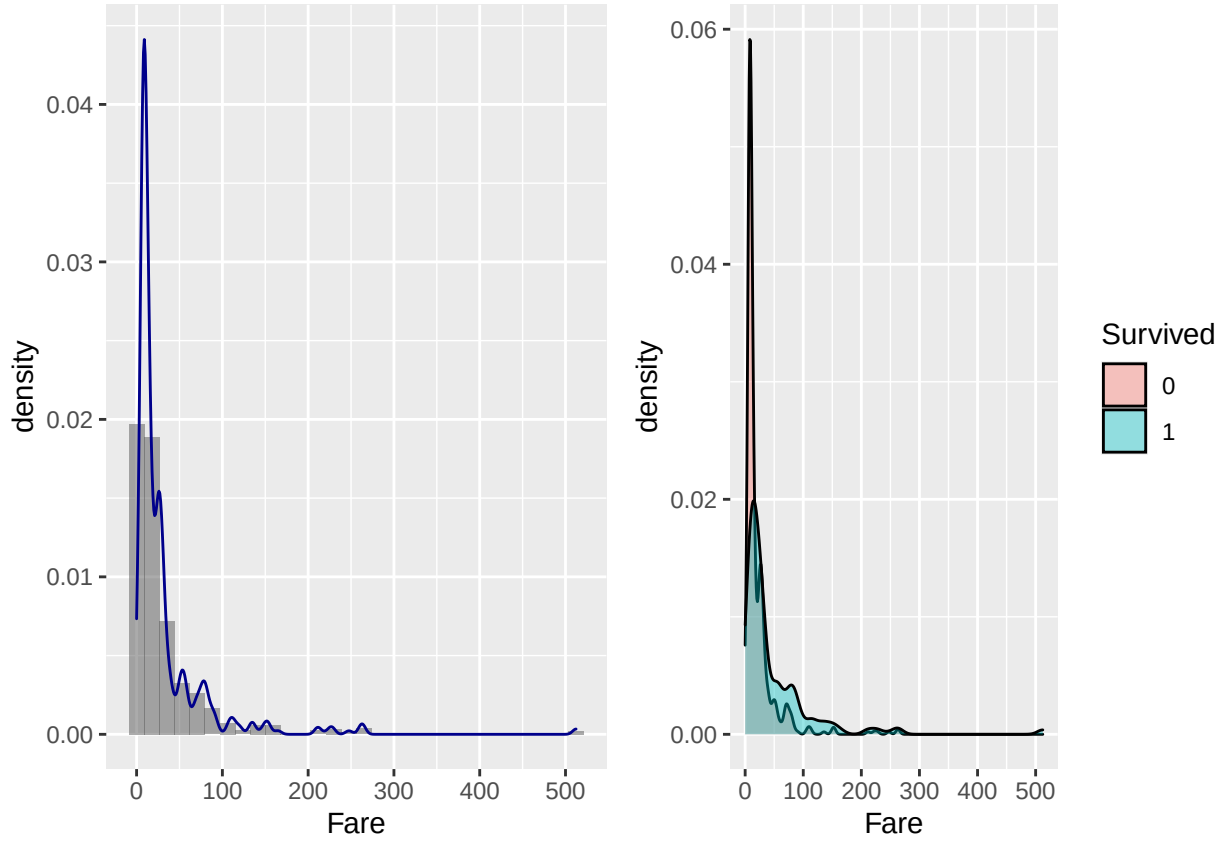
```
##
##  Shapiro-Wilk normality test
##
## data:  new.titanic.train$Age
## W = 0.98264, p-value = 8.517e-09
```

La salida anterior tiene un valor p 8.517e-09 para la prueba, dado que esto es menor que el nivel de significancia de 5% se debe rechazar la hipótesis nula que las edades de los pasajeros en la variable "Age" se distribuyen normalmente.

Con la variable “Fare” se grafica el histograma y su densidad para saber si los datos tienen una distribución normal y su relación con la “variable” Survived para ver la condición de supervivencia.

```
fare.density.plot <- ggplot(new.titanic.train, aes(x = Fare))+
  geom_histogram(aes(y = ..density..), alpha = 0.5, position = "identity")+
  geom_density(color = "darkblue", alpha = 0.2)

fare.survived.density.plot <- ggplot(new.titanic.train, aes(x = Fare, fill = Survived))+
  geom_density(alpha = 0.4)

grid.arrange(fare.density.plot, fare.survived.density.plot, ncol = 2)
```

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

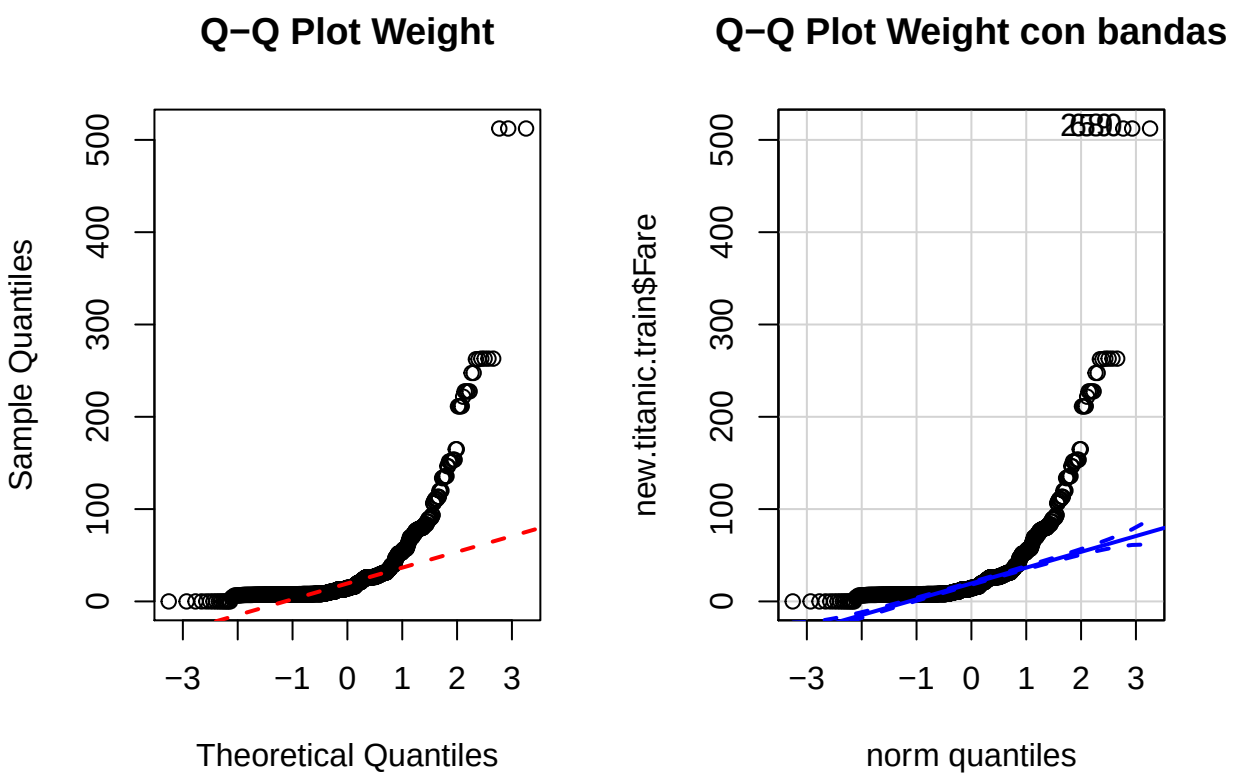


En el gráfico se observa:

- Que las densidades son asimétricas.
- La mayor densidad se encuentra en el intervalo de 0 a 50 (tarifa de boleto).
- Hay un sesgo hacia las tarifas de menor valor y unos pocos de mayor valor, esto hace que se desconfíe de la normalidad de los datos.

Con la función qqplot() se organiza los puntos respecto a la línea de referencia y bandas para los puntos en la zona de normalidad.

```
par(mfrow = c(1,2))
qqnorm(new.titanic.train$Fare, main = "Q-Q Plot Weight")
qqline(new.titanic.train$Fare, col="red", lwd = 2, lty = 2)
qqPlot(new.titanic.train$Fare, main = "Q-Q Plot Weight con bandas")
```




```
## [1] 259 680
```

En el gráfico se observa que la densidad de tarifa del billete no tiene simetría y posee sesgo a los lados.

```
shapiro.test(new.titanic.train$Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  new.titanic.train$Fare
## W = 0.52189, p-value < 2.2e-16
```

Por medio de las pruebas de normalidad el valor p de ambas muestras es menor al nivel de significancia de 5%, por tanto se puede rechazar la hipótesis nula que la tarifa del billete distribuye normalmente.

4.2.2 Homogeneidad de la varianza

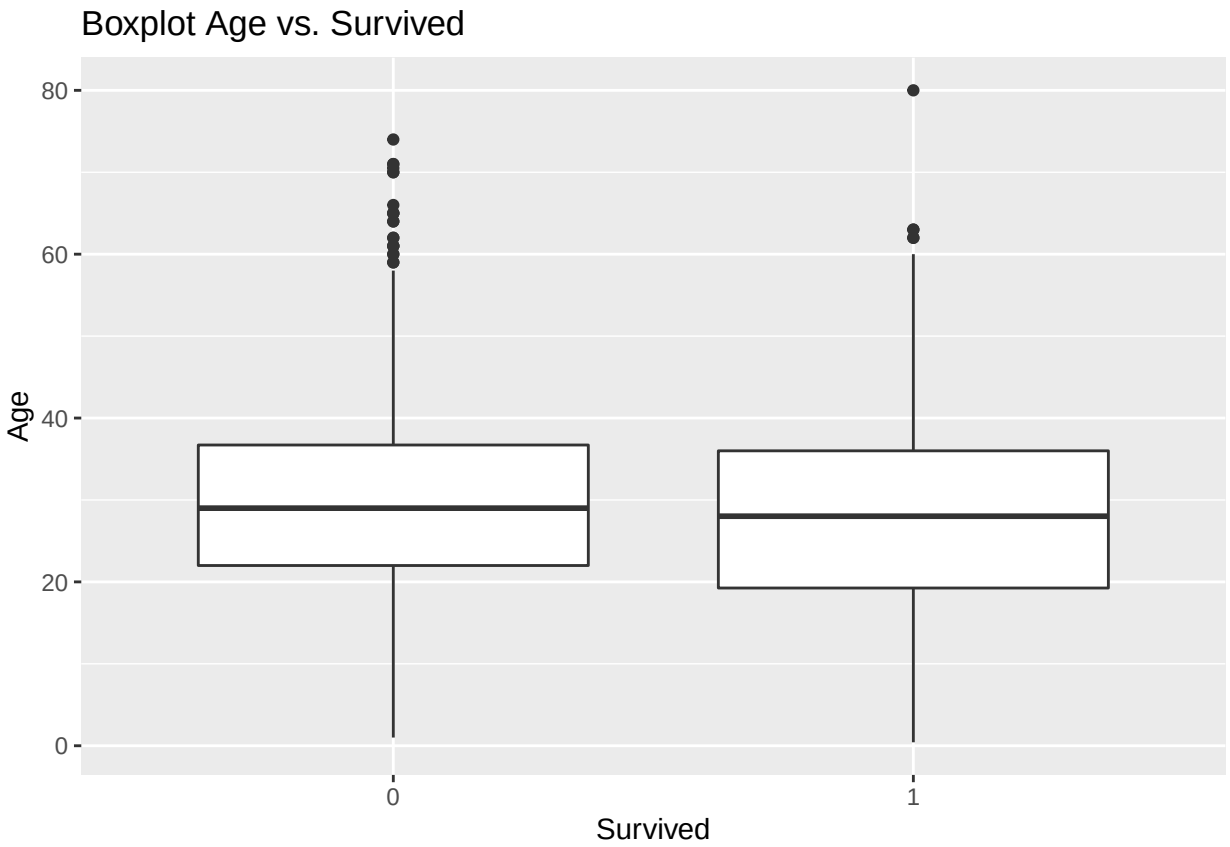
Como no tenemos la certeza de normalidad con las variables antes revisadas, para el análisis de la homgeneidad de varianza emplearemos el test no paramétrico Fligner-Killeen que se basa en la mediana. La hipótesis sería la siguiente:

Hipótesis nula (H_0): Las varianzas son iguales.

Hipótesis alternativa (H_1): Al menos dos de ellos difieren.

Aplicamos el supuesto de homogeneidad para la variable "Age".

```
ggplot(new.titanic.train, aes(x=Survived, y=Age))+
  geom_boxplot()+ggtitle("Boxplot Age vs. Survived")
```



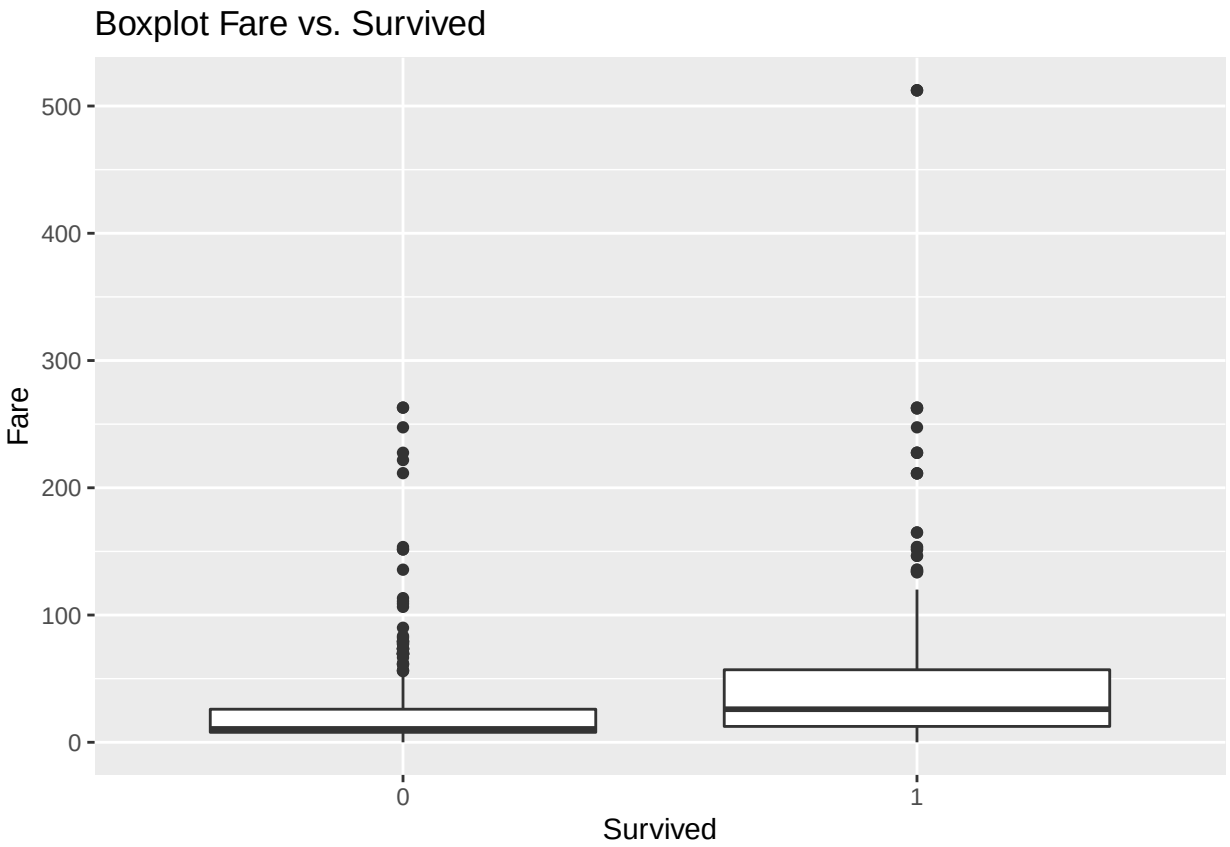

```
# Prueba de Figner-Killeen de homogeneidad de varianzas
fligner.test(Age ~ Survived, data = new.titanic.train)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 6.3187, df = 1, p-value = 0.01195
```

El resultado de la función genera un valor p = 0.01195, siendo menor que el nivel de significancia de 5% rechazando la hipótesis nula, no existe homogeneidad de varianzas entre Age y Survived.

Ahora revisamos el supuesto de homogeneidad para la variable “Fare”.

```
ggplot(new.titanic.train, aes(x=Survived, y=Fare))+
  geom_boxplot()+ggtitle("Boxplot Fare vs. Survived")
```




```
# Prueba de Figner-Killeen de homogeneidad de varianzas
fligner.test(Fare ~ Survived, data = new.titanic.train)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by Survived
## Fligner-Killeen:med chi-squared = 96.253, df = 1, p-value < 2.2e-16
```

Se obtiene valor p < 2.2e-16, el cual es menor al nivel de significancia de 5% rechazando la hipótesis nula, es decir, no hay homogeneidad de varianzas entre "Fare" y "Survived".

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

Resumimos una estadística descriptiva del conjunto de datos tratado. Para las variables numéricas revisamos el valor mínimo, máximo, la media, mediana y los tres cuartiles y no se observa novedad. Se destaca que el valor de la media en "Age" es bastante similar a la mediana. 29.68 y 28.97 respectivamente.

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

```
##   PassengerId    Survived Pclass      Name              Sex
##  Min.   :  1.0   0:549    1:216   Length:891         female:314
##  1st Qu.:223.5   1:342    2:184   Class :character   male  :577
##  Median :446.0            3:491   Mode  :character
```

```
##  Mean   :446.0
##  3rd Qu.:668.5
##  Max.   :891.0
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:21.03   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.97   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.68   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:36.71   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##
##       Fare           Cabin           Embarked Designation    FamilySize
##  Min.   :  0.00   Length:891         C:168    Master: 40   1      :537
##  1st Qu.:  7.91   Class :character   Q: 77    Miss  :185   2      :161
##  Median : 14.45   Mode  :character   S:646    Mr    :517   3      :102
##  Mean   : 32.20                               Mrs   :126   4      : 29
##  3rd Qu.: 31.00                               Other : 23   6      : 22
##  Max.   :512.33                                            5      : 15
##                                                            (Other): 25
##  FamilyGroup AgeForGroup
##  Alone:537   Adult:763
##  Large: 62   Child:128
##  Small:292
##
##
##
##
```

Se emplea la función str() para visualizar el tipo de variable de cada columna y una parte de su contenido. Se constatan variables categóricas.

```
str(new.titanic.train)
```

```
## 'data.frame':    891 obs. of  16 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 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 ...
##  $ 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  NA "C85" NA "C123" ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  $ Designation: Factor w/ 5 levels "Master","Miss",..: 3 4 2 4 3 3 3 1 4 4 ...
##  $ FamilySize : Factor w/ 9 levels "1","2","3","4",..: 2 2 1 2 1 1 1 5 3 2 ...
##  $ FamilyGroup: Factor w/ 3 levels "Alone","Large",..: 3 3 1 3 1 1 1 2 3 3 ...
##  $ AgeForGroup: Factor w/ 2 levels "Adult","Child": 1 1 1 1 1 1 1 2 1 2 ...
```

Ahora aplicaremos la función describe() de la librería psych para obtener una variedad de resultados estadísticos descriptivos a la vez, tales como: nombre del atributo, número de atributo, cantidad de datos validados, media, desviación estándar, mediana, media recortada, desviación media absoluta, valor mínimo y máximo, rango (valor máximo - valor mínimo), coeficiente asimetria (skew), curtosis y error estándar de la media (se)

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

```r
describe(new.titanic.train)
```

```
##              vars   n   mean     sd median trimmed    mad  min    max  range
## PassengerId     1 891 446.00 257.35 446.00  446.00 330.62 1.00 891.00 890.00
## Survived*       2 891   1.38   0.49   1.00    1.35   0.00 1.00   2.00   1.00
## Pclass*         3 891   2.31   0.84   3.00    2.39   0.00 1.00   3.00   2.00
## Name*           4 891 446.00 257.35 446.00  446.00 330.62 1.00 891.00 890.00
## Sex*            5 891   1.65   0.48   2.00    1.68   0.00 1.00   2.00   1.00
## Age             6 891  29.68  13.56  28.97   29.32  11.77 0.42  80.00  79.58
## SibSp           7 891   0.52   1.10   0.00    0.27   0.00 0.00   8.00   8.00
## Parch           8 891   0.38   0.81   0.00    0.18   0.00 0.00   6.00   6.00
## Ticket*         9 891 339.52 200.83 338.00  339.65 268.35 1.00 681.00 680.00
## Fare           10 891  32.20  49.69  14.45   21.38  10.24 0.00 512.33 512.33
## Cabin*         11 204  77.00  42.23  76.00   77.09  54.11 1.00 147.00 146.00
## Embarked*      12 891   2.54   0.79   3.00    2.67   0.00 1.00   3.00   2.00
## Designation*   13 891   2.90   0.79   3.00    2.89   0.00 1.00   5.00   4.00
## FamilySize*    14 891   1.89   1.53   1.00    1.52   0.00 1.00   9.00   8.00
## FamilyGroup*   15 891   1.73   0.93   1.00    1.66   0.00 1.00   3.00   2.00
## AgeForGroup*   16 891   1.14   0.35   1.00    1.05   0.00 1.00   2.00   1.00
##                skew kurtosis   se
## PassengerId    0.00    -1.20 8.62
## Survived*      0.48    -1.77 0.02
## Pclass*       -0.63    -1.28 0.03
## Name*          0.00    -1.20 8.62
## Sex*          -0.62    -1.62 0.02
## Age            0.37     0.45 0.45
## SibSp          3.68    17.73 0.04
## Parch          2.74     9.69 0.03
## Ticket*        0.00    -1.28 6.73
## Fare           4.77    33.12 1.66
## Cabin*         0.00    -1.19 2.96
## Embarked*     -1.26    -0.22 0.03
## Designation* -0.05     0.70 0.03
## FamilySize*    2.34     5.79 0.05
## FamilyGroup*   0.57    -1.59 0.03
## AgeForGroup*   2.03     2.12 0.01
```

Con el resumen de datos expuesto anteriormente destacamos:

- Existen valores perdidos en la variable "Cabin" siendo correcto porque procederemos a descartarla en los métodos a emplear.
- El valor de la media en todas las variables numéricas es similar a la mediana, excepto en el atributo "Fare" dado que su media es 32.20 y mediana 14.45.
- Para las variables "Pclass", "Sex", y "Embarked", el coeficiente de simetría es negativo, es decir, es una variable de tipo asimétrica hacia la izquierda de la distribución de casos.

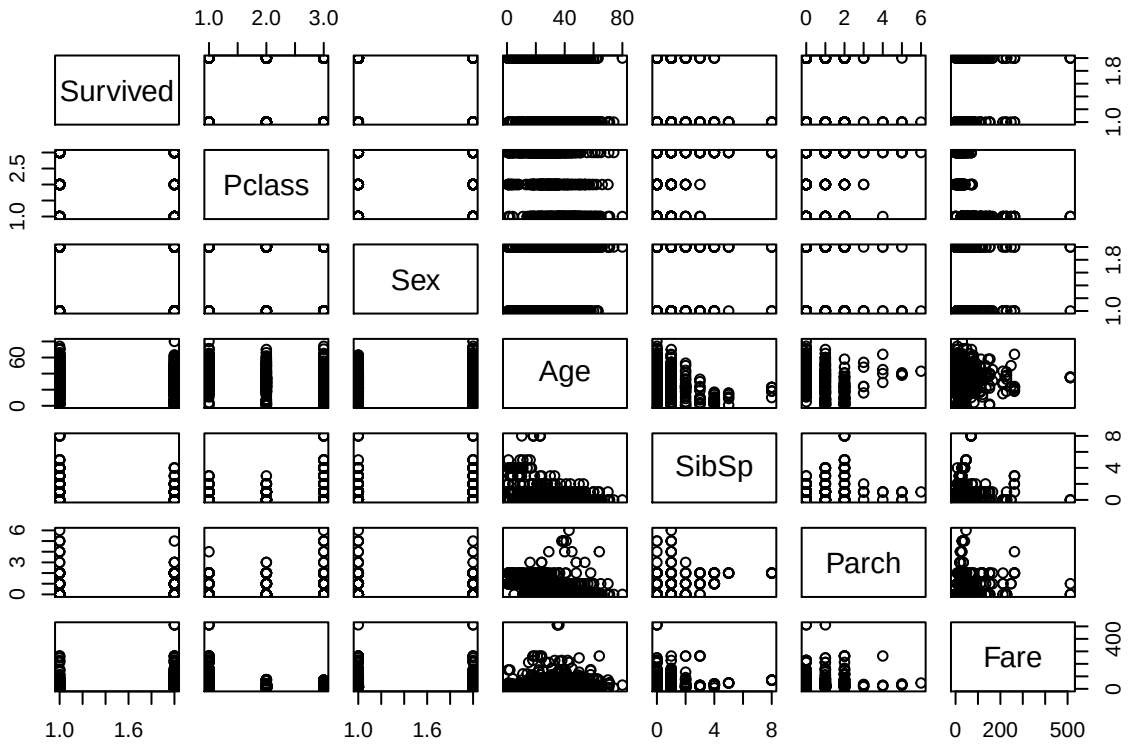
En este punto se desconoce cuán relacionadas están las variables cuantitativas intervinientes, por lo que es importante descubrir la covarianza y correlación para resumir su relación lineal.

```
# Seleccionar las variables numéricas
titanic.numeric.data <- subset(new.titanic.train,
 select = c("Survived","Pclass","Sex","Age","SibSp","Parch","Fare"))
```

Se estudia la correlación de cada variable mediante un diagrama de dispersión múltiple, es la mejor manera de evaluar la linealidad entre dos variables.

Creo un gráfico de dispersión para revisar la relación que existe entre cada par de variables: "Survived" con "Pclass", "Sex", "Age", "SibSp", "Parch", "Fare". Así determinamos si existe relación lineal con la variable respuesta o colineidad entre variables.

```
# Mostrar matriz gráfica de dispersión entre variables
pairs(titanic.numeric.data)
```



Analizando la relación entre las variables numéricas tenemos:

```
cor(titanic.numeric.data[,c("Age","SibSp","Parch","Fare")])
```

```
##               Age      SibSp      Parch       Fare
## Age    1.00000000 -0.2943762 -0.2079774 0.09882429
## SibSp -0.29437623  1.0000000  0.4148377 0.15965104
## Parch -0.20797741  0.4148377  1.0000000 0.21622494
## Fare   0.09882429  0.1596510  0.2162249 1.00000000
```

- El coeficiente devuelve un valor entre -1 y 1.
- La correlación es tanto más fuerte cuanto más se aproxime a 1.
- La correlación negativa es tanto más fuerte cuanto más se aproxime a -1.

El análisis gráfico y los valores de correlación para las variables numéricas muestran una fuerte relación positiva en “Age” y “Fare”.

En tanto que, la covarianza es el estadístico que indica el sentido de los valores de muestras.

- El valor positivo muestra que ambas variables varían en la misma dirección.
- El valor negativo muestra que varían en la dirección opuesta.

En R se calcula con la función cov().

```
cov(titanic.numeric.data[,c("Age","Fare")])
```

```
##            Age     Fare
## Age  183.9783   66.611
## Fare  66.6110 2469.437
```

La covarianza sólo informa sobre la dirección. En complemento, el coeficiente de correlación explica sobre el cambio en una variable e indica cuánto cambió de proporción en la segunda variable.

Los resultados de correlación y covarianza se resumen a continuación:

- El diagrama de dispersión tiene cierto patrón lineal, sí se alcanza a ver la dirección de los puntos, pero con fuerza de relación baja, se destacan las variables Age y Fare.
- El valor de covarianza de las variables Age y Fare son positivos y se interpreta que cambian en la misma dirección.
- La cuantificación del coeficiente de correlación establece baja relación entre las variables Age y Fare.

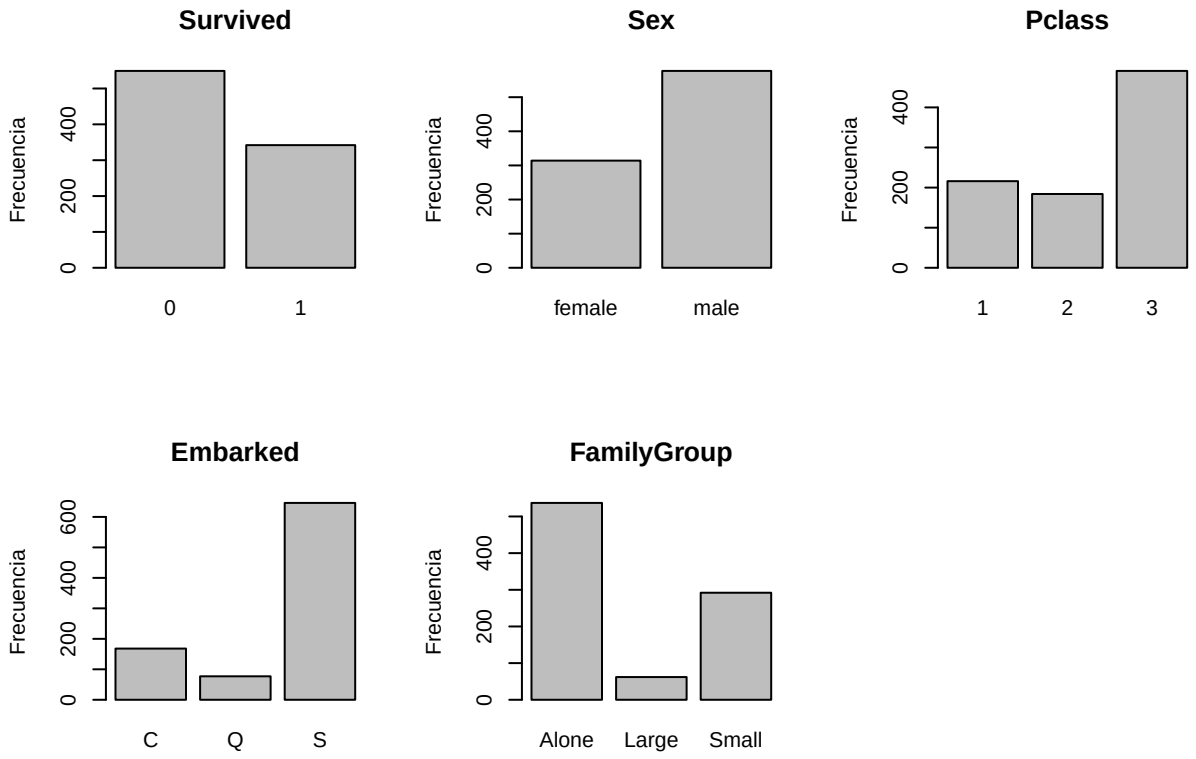
5 Representación de los resultados a partir de tablas y gráficas.

Para las variables “Survived”, “Sex”, “Pclass”, “Embarked” y “FamilyGroup” como son factores creamos un gráfico de barras.

- En el gráfico de Survived la mayor concentración está en la categoría 0 (no sobrevivieron).
- En el gráfico de Sex se observa que el género se concentra mayormente en hombres.
- En el gráfico de Pclass se evidencia que la clase de boleto se representa significativamente en la clase 3.
- En el gráfico de Embarked se evidencia que el puerto de embarque con mayor registro fue en S=Southampton.
- En el gráfico de FamilyGroup se denota que la mayor concentración está en pasajeros a bordo sin familiares.

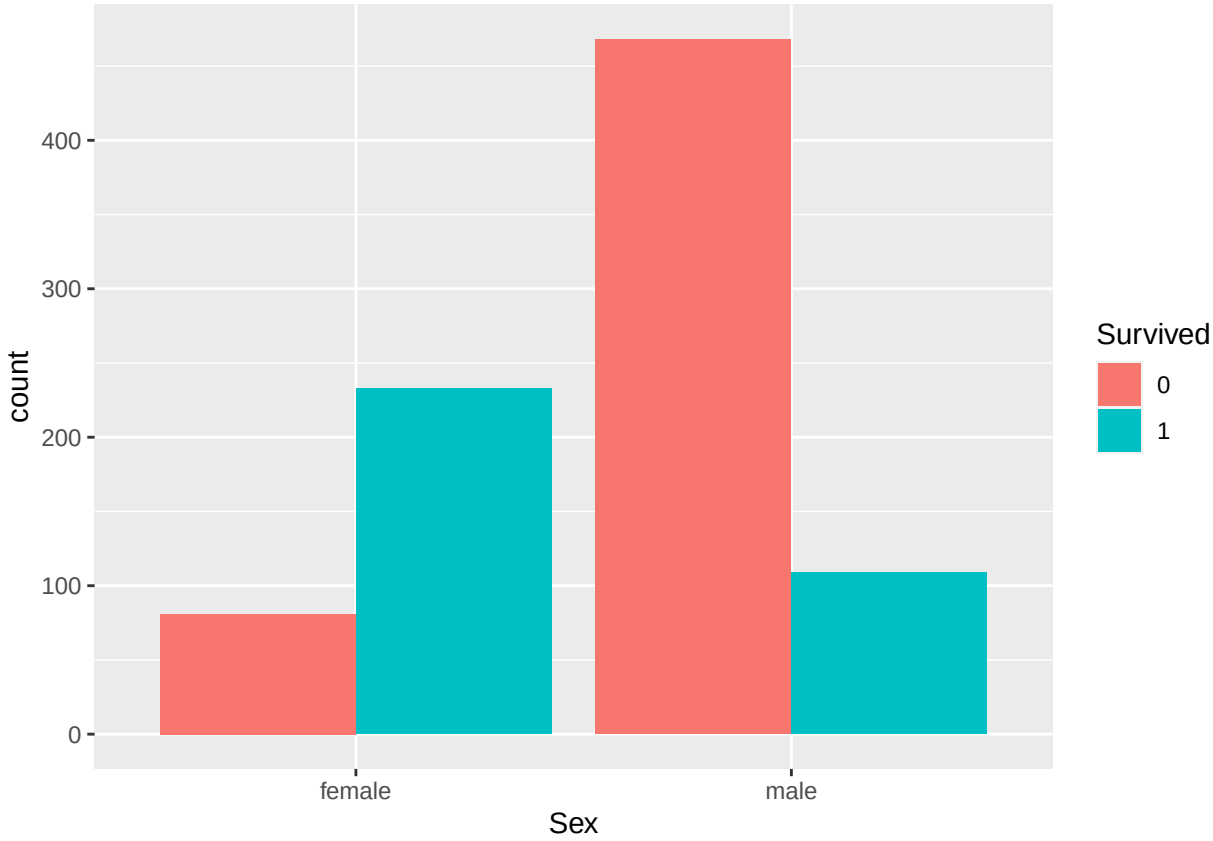
```
par(mfrow=c(2, 3))

Survived1 <- table(new.titanic.train$Survived);
barplot(Survived1, main="Survived", ylab="Frecuencia")

Sex1 <- table(new.titanic.train$Sex);
barplot(Sex1, main="Sex", ylab="Frecuencia")

Pclass1 <- table(new.titanic.train$Pclass);
barplot(Pclass1, main="Pclass", ylab="Frecuencia")

Embarked1 <- table(new.titanic.train$Embarked);
barplot(Embarked1, main="Embarked", ylab="Frecuencia")
```

```
FamilyGroup1 <- table(new.titanic.train$FamilyGroup);
barplot(FamilyGroup1, main="FamilyGroup", ylab="Frecuencia")

par(mfrow=c(1, 1))
```



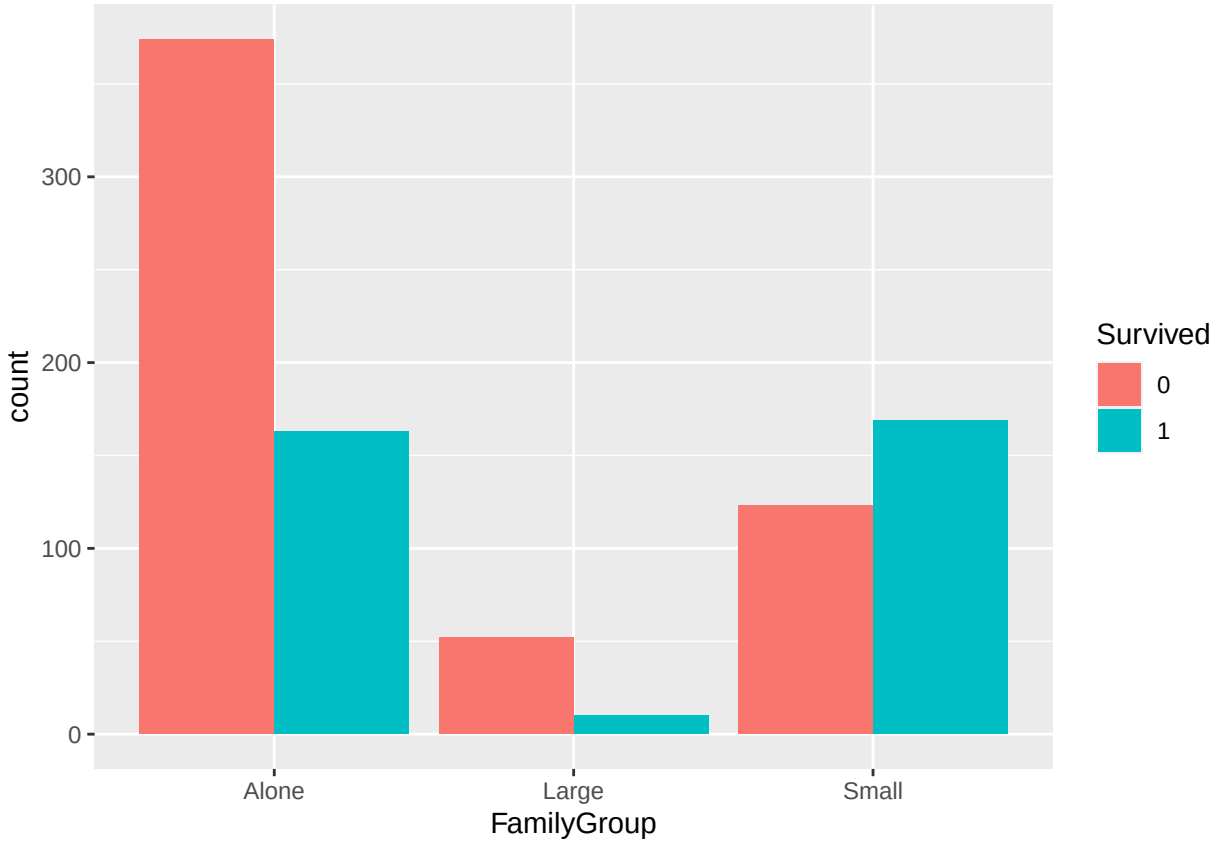
Mediante el siguiente gráfico evaluamos por sexo quiénes sobrevivieron y se denota que son las mujeres.

```
# Visualizamos la relación entre las variables "sex" y "survived":
ggplot(data=new.titanic.train, aes(x=Sex,fill=Survived))+
  geom_bar(position = 'dodge')
```



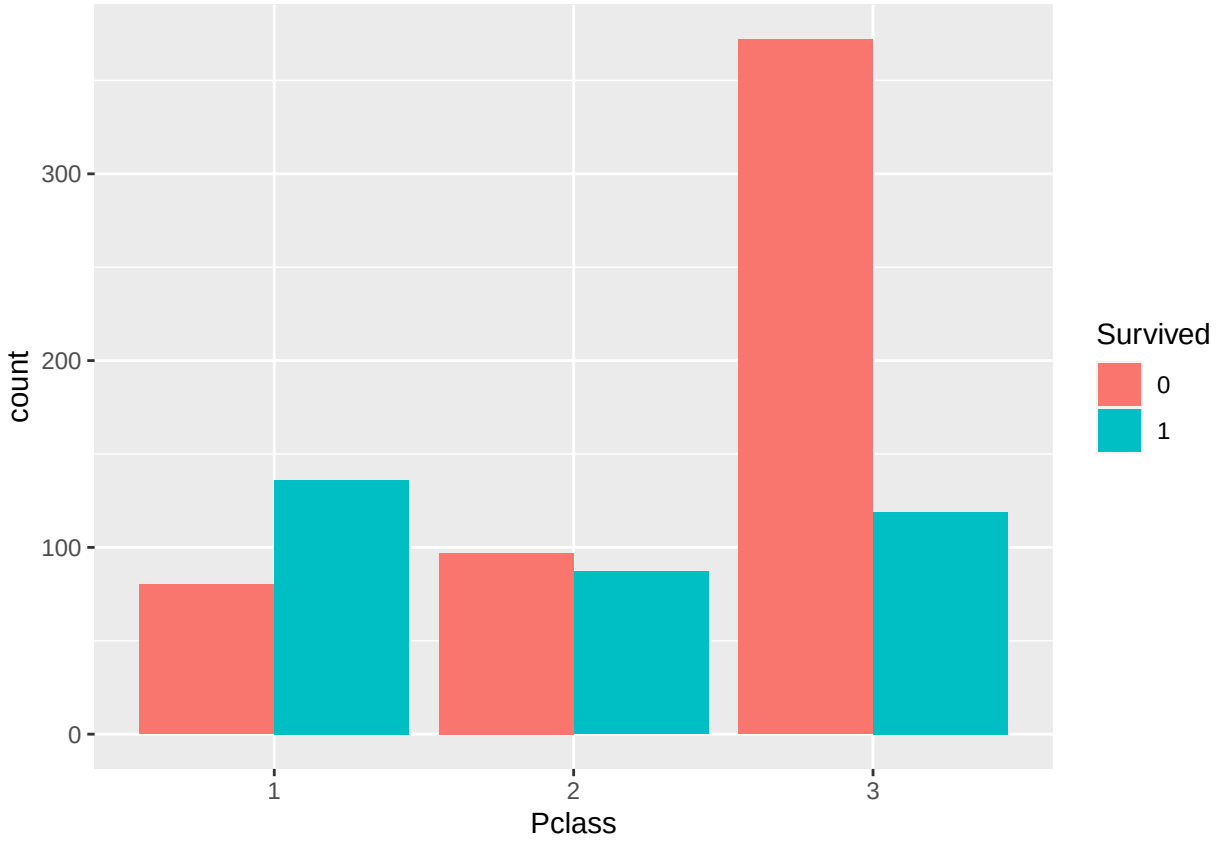
Los pasajeros sin familiares a bordo del barco tienen mayor probabilidad de supervivencia.

```
# Visualizamos la relación entre las variables "FamilyGroup" y "survived":
ggplot(data=new.titanic.train,aes(x=FamilyGroup,fill=Survived))+
  geom_bar(position = 'dodge')
```



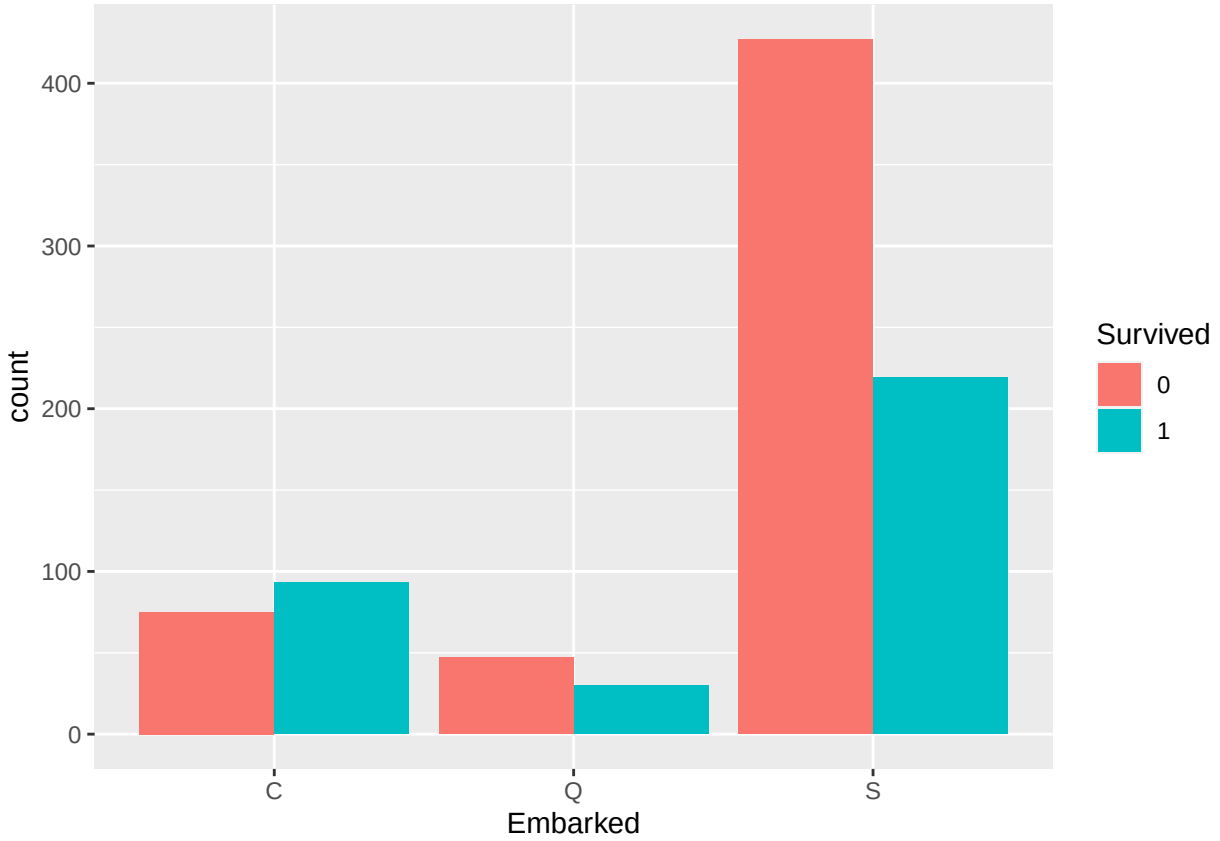
Si analizamos la supervivencia en razón de la clase de boleto del pasajero focalizamos que en Pclass = 1 hay mayor número de sobrevivientes.

```
# Visualizamos la relación entre las variables "Pclass" y "survived":
ggplot(data=new.titanic.train,aes(x=Pclass,fill=Survived))+geom_bar(position = 'dodge')
```



La supervivencia según el puerto de embarque es mayor en Southampton (S).

```
# Visualizamos la relación entre las variables "Embarked" y "survived":
ggplot(data=new.titanic.train,aes(x=Embarked,fill=Survived))+geom_bar(position = 'dodge')
```



El objetivo del análisis es determinar las relaciones enre las variables disponibles y la supervivencia de los pasajeros, entonces se pasará a construir los modelos de predicción.

5.1 Modelo Regresión Logística

Cargamos las librerías necesarias para la creación del modelo.

```
# Cargar liberías necesarias
library(caret)
library(scales)
library(vip)
library(dplyr)
library(pROC)
library(WVPlots)
library(forcats)
library(recipes)
```

Estimamos un modelo de regresión logística con la variable dependiente Survived y los regresores Pclass, Sex, Age, AgeForGroup, Fare, Embarked, Designation y FamilyGroup. La variable Survived es una variable dicotómica, que toma el valor 0 cuando el pasajero no sobrevive y 1 cuando el pasajero sobrevive.

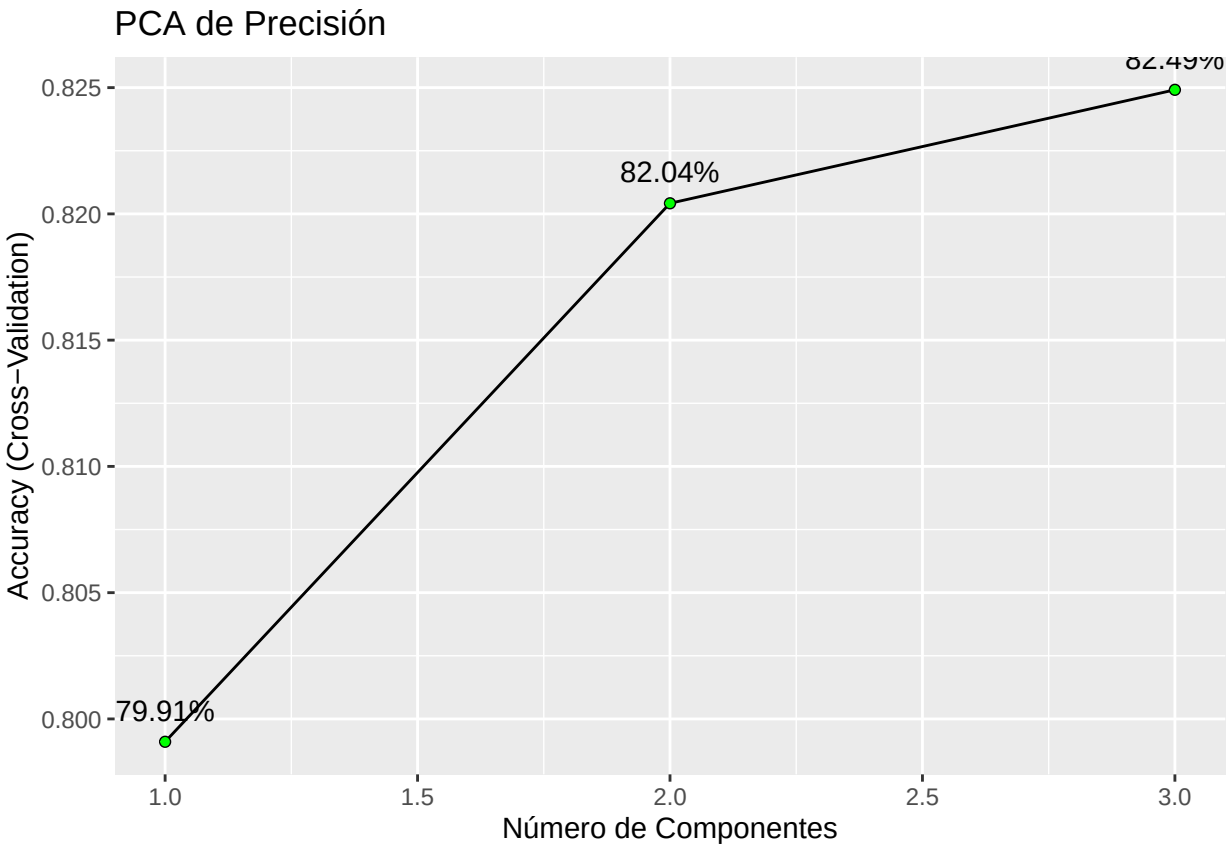
Se aplica validación cruzada. El método a emplear es el de regresión de mínimos cuadrados parciales (PLS) basado en la covarianza y relacionado con la regresión de componentes principales.

```
titanic_rl <- new.titanic.train[,c("Survived","Pclass","Sex","Age","AgeForGroup","Fare",
 "Embarked","Designation","FamilyGroup")]

set.seed(123)
```

```
ctrl <- trainControl(method = "cv", number = 10)

model_pls <- train(
  Survived ~ .,
  data = titanic_rl,
  method = "pls",
  family = "binomial",
  preProcess = c("zv","center", "scale"),
  trControl = ctrl,
  tuneLenght = 8
)
```

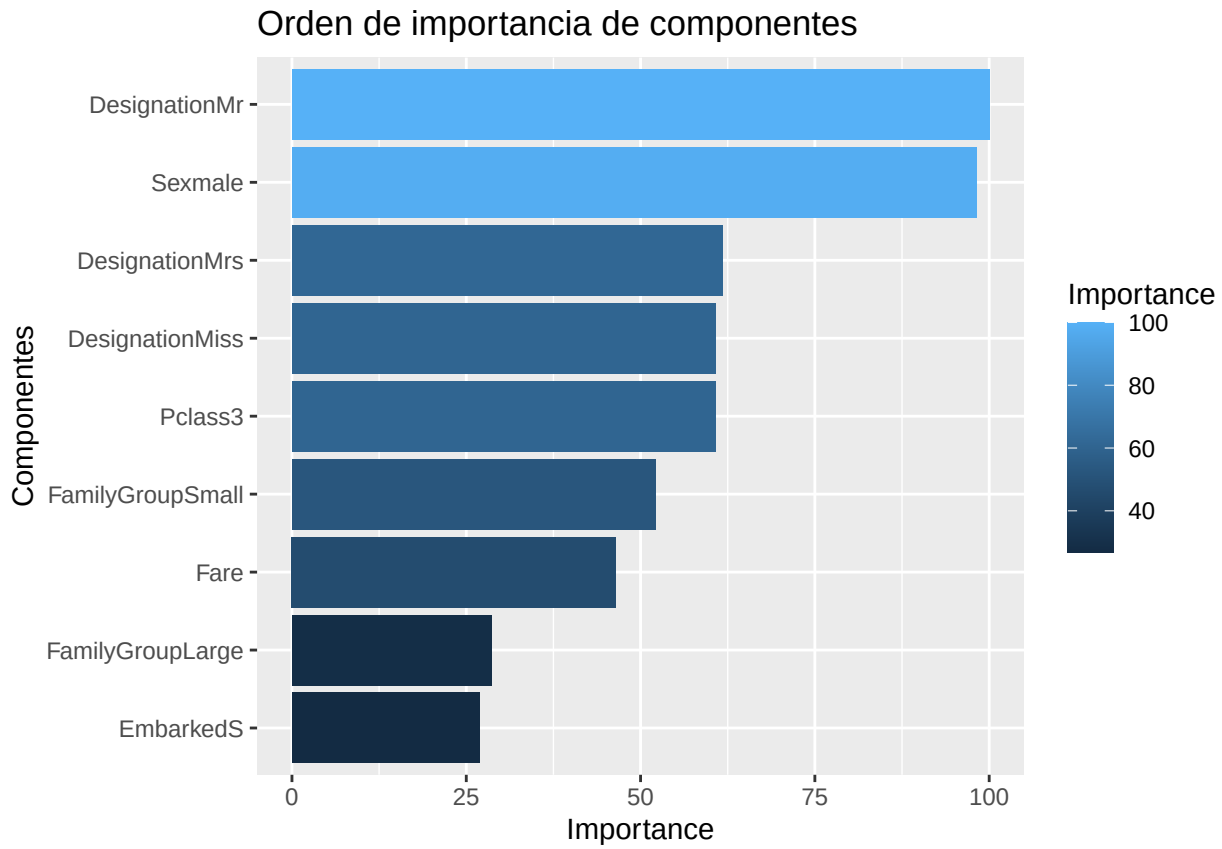
Graficamos los principales componentes de análisis de precisión notando que tenemos la precisión más alta es de 82.49%

```
ggplot(model_pls) +
  geom_point(color = "green", size = 1) +
  geom_text(aes(x = ncomp, y = Accuracy, label = percent(Accuracy)),
            vjust = -1) +
  labs(
    title = "PCA de Precisión",
    x = "Número de Componentes"
  )
```



Determinando las variables más importantes tenemos: Sex (male), Pclass (3) y Fare.

```r
vip1 <- vip(model_pls, num_features = 9, method = "model")

vip1$data %>%
  ggplot(aes(x = Importance, y = reorder(Variable, Importance), fill = Importance)) +
    geom_col() +
    labs(
      title = "Orden de importancia de componentes",
      y = "Componentes"
    )
```



Como vimos anteriormente, la precisión del modelo es alta (82.60%) y al crear la matriz de confusión obtenemos alta especificidad (88.71%) lo que significa que el modelo fue preciso en la predicción de verdaderos negativos. En otras palabras, tiene un buen desempeño para predecir quién no sobrevivió.

El 83.97% de todas las muertes predichas son correctas. El 80.06% de sobrevivientes predichas son correctas.

```r
titanic_rl$survival_pred <- predict(model_pls, titanic_rl)
model_prob <- predict(model_pls, newdata = titanic_rl, type = "prob")

titanic_rl <- titanic_rl %>%
  mutate(survival_prob = model_prob[,2])

cfMatrix <- confusionMatrix(
  data = relevel(titanic_rl$survival_pred, ref = "1"),
  reference = relevel(titanic_rl$Survived, ref = "1")
)
```

```
cfMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   0
##          1 249  62
##          0  93 487
##
##                Accuracy : 0.826
##                  95% CI : (0.7995, 0.8504)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6258
##
##  Mcnemar's Test P-Value : 0.01597
##
##             Sensitivity : 0.7281
##             Specificity : 0.8871
##          Pos Pred Value : 0.8006
##          Neg Pred Value : 0.8397
##              Prevalence : 0.3838
##          Detection Rate : 0.2795
##    Detection Prevalence : 0.3490
##       Balanced Accuracy : 0.8076
##
##        'Positive' Class : 1
##
```

Un método para evaluar clasificadores alternativo a la métrica expuesta es la curva ROC (Receiver Operating Characteristic). La curva ROC es una representación gráfica del rendimiento del clasificador que muestra la distribución de las fracciones de verdaderos positivos y de falsos positivos.

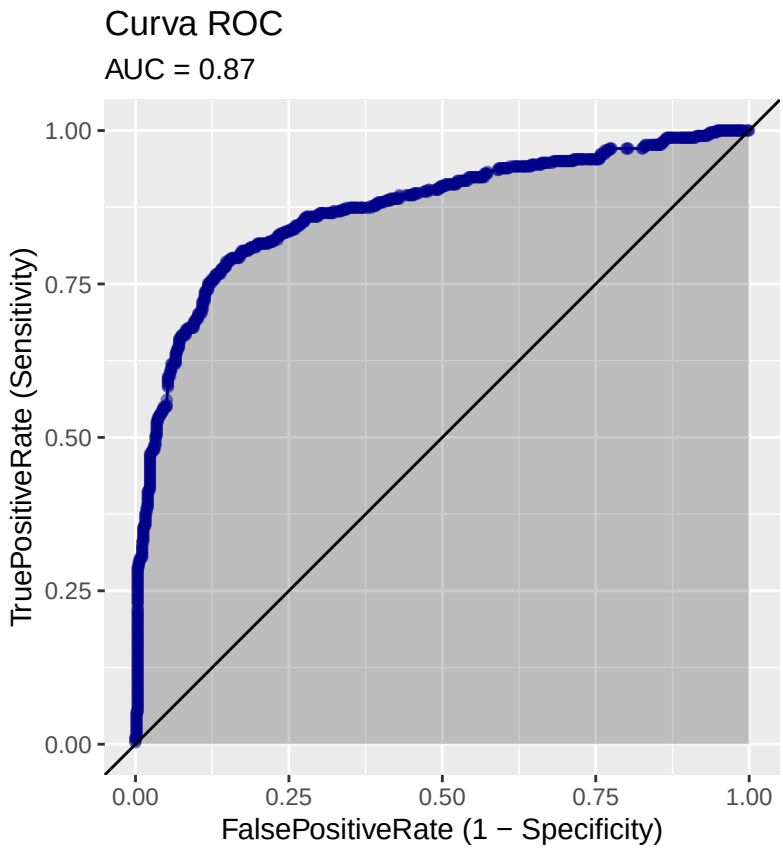
La curva ROC nos ayudará a comprender cómo le fue al modelo en comparación con simplemente adivinar los resultados al azar.

```
ROC <- roc(titanic_rl$Survived, as.numeric(titanic_rl$survival_pred))
```

```
## Setting levels: control = 0, case = 1
```

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

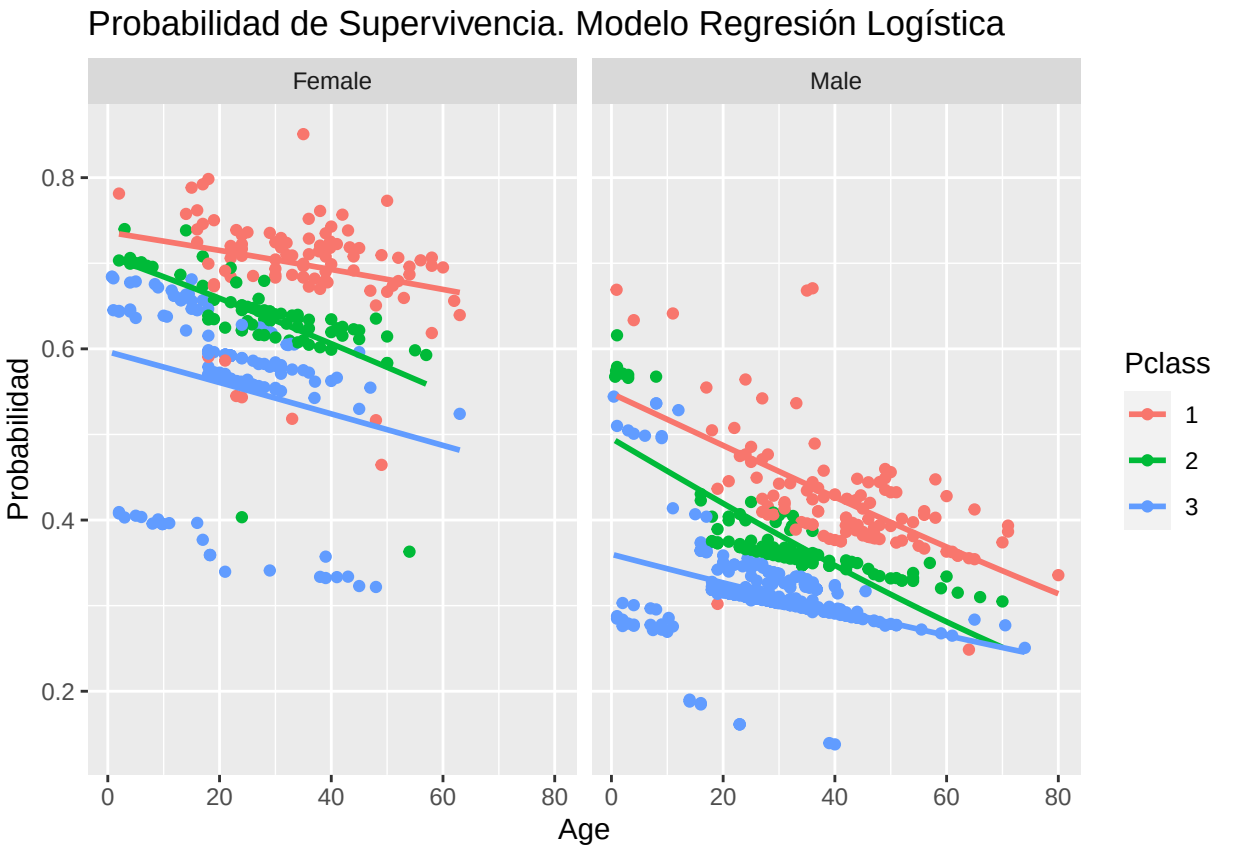
```
titanic_train2 <- titanic_rl %>%
  mutate(Survived = as.numeric(as.character(Survived)),
         survival_prob = as.numeric(as.character(survival_prob)))

ROCPlot(titanic_train2, "survival_prob", "Survived",
        truthTarget = TRUE, title = "Curva ROC") +
  labs(
    title = "Curva ROC"
  )
```



Un área bajo la curva de 0.87 significa que el modelo funcionó bien. Significa que la tasa de verdaderos positivos crece a un ritmo más rápido que la tasa de falsos positivos.

Al dibujar la probabilidad de supervivencia, es notorio ver que a mayor edad del pasajero, menor es la probabilidad de supervivencia. También es evidente la diferencia de supervivencia según el sexo y la clase de boleto adquiridad. Así por ejemplo, las mujeres tienen mayor grado de supervivencia que los hombres y respecto al boleto del pasajero, la probabilidad de sobrevivir es mayor si ha adquirido un boleto de clase alta.

```
titanic_rl %>%
  mutate(Sex = fct_recode(Sex , "Male" = "male", "Female" = "female")) %>%
  ggplot(aes(x = Age, y = survival_prob, color = Pclass)) +
    geom_point() +
    geom_smooth(method = "glm", method.args = list(family = "binomial"),
      se = FALSE) +
    facet_wrap(~Sex) +
    labs(
      title = "Probabilidad de Supervivencia. Modelo Regresión Logística",
      y = "Probabilidad",
      color = "Pclass"
    )
```



Procedemos a estimar la supervivencia con el modelo creado.

```
titanic_train_recipe <- new.titanic.train %>%
  mutate(Survived = ifelse(Survived =="1",0,1)) %>%
  select(-c("Name", "Ticket", "Cabin","SibSp", "Parch"))

test_recipe <- recipe(Survived ~ ., data = titanic_train_recipe)

prepare <- prep(test_recipe, training = titanic_train_recipe, strings_as_factors = TRUE)

titanic_rl_test <- new.titanic.test %>%
  mutate(Survived = as.integer(0)) %>%
  select(c("PassengerId","Survived", "Pclass","Sex","Age","AgeForGroup","Fare","Embarked",
 "Designation","FamilyGroup"))

baked_rl_test <- bake(prepare, new_data = titanic_rl_test)

baked_rl_test <- baked_rl_test %>%
  mutate(Pclass = factor(Pclass),
         Sex = factor(Sex))

baked_rl_test$prediction_rl <- predict(model_pls, baked_rl_test)

baked_rl_test %>%  select(PassengerId,prediction_rl)
```

```
## # A tibble: 418 x 2
##    PassengerId prediction_rl
```

```
##          <int> <fct>
##  1         892 0
##  2         893 1
##  3         894 0
##  4         895 0
##  5         896 1
##  6         897 0
##  7         898 1
##  8         899 0
##  9         900 1
## 10         901 0
## # ... with 408 more rows
```

El resultado en el conjunto test aplicando el modelo de regresión es que de 418 pasajeros, 163 personas sobrevivieron.

```
summary(baked_rl_test$prediction_rl)
```

```
##   0   1
## 255 163
```

5.2 Modelo Random Forest

El modelo Random Forest se conforma por un conjunto de árboles de decisión individuales, cada árbol se entrena con unos datos ligeramente distintos.

En cada árbol individual, las observaciones se distribuyen por nodos generando la estructura del árbol hasta alcanzar un nodo terminal. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo.

El método asociado a random forest para validación cruzada es ranger.

```
titanic_rf <- new.titanic.train[,c("Survived","Pclass","Sex","Age","AgeForGroup",
  "Fare","Embarked","Designation","FamilyGroup")]

set.seed(123)

response <- "Survived"
predictors <- setdiff(names(titanic_rf), response)
n_features <- length(predictors)

ctrl <- trainControl(method = "cv", number = 10)

rf_grid <- expand.grid(
  mtry = floor(n_features * c(.25,.33,.50,.66,.75)),
  min.node.size = c(1,3,5,7,10),
  splitrule = c("gini", "extratrees")
)


model_rf <- train(
  Survived~.,
  data=titanic_rf,
  method='ranger',
  metric='Accuracy',
  tuneGrid = rf_grid,
  preProcess = c("zv","center", "scale"),
```

```
  trControl=ctrl)

model_rf
```

```
## Random Forest
##
## 891 samples
##   8 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (14), scaled (14)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 802, 802, 801, 801, 802, 802, ...
## Resampling results across tuning parameters:
##
##   mtry  min.node.size  splitrule   Accuracy   Kappa
##   2      1             gini        0.8305439  0.6331257
##   2      1             extratrees  0.8294331  0.6309461
##   2      3             gini        0.8294078  0.6305605
##   2      3             extratrees  0.8305442  0.6331423
##   2      5             gini        0.8316678  0.6345798
##   2      5             extratrees  0.8316678  0.6359861
##   2      7             gini        0.8350386  0.6421051
##   2      7             extratrees  0.8328039  0.6393422
##   2     10             gini        0.8283220  0.6286436
##   2     10             extratrees  0.8305442  0.6335223
##   4      1             gini        0.8384352  0.6483527
##   4      1             extratrees  0.8305570  0.6283005
##   4      3             gini        0.8395588  0.6506503
##   4      3             extratrees  0.8328169  0.6335054
##   4      5             gini        0.8384227  0.6475565
##   4      5             extratrees  0.8316806  0.6306266
##   4      7             gini        0.8395588  0.6501281
##   4      7             extratrees  0.8350516  0.6390915
##   4     10             gini        0.8395588  0.6504730
##   4     10             extratrees  0.8361752  0.6404313
##   5      1             gini        0.8418063  0.6559709
##   5      1             extratrees  0.8294461  0.6261321
##   5      3             gini        0.8429168  0.6591550
##   5      3             extratrees  0.8316933  0.6303868
##   5      5             gini        0.8429043  0.6597051
##   5      5             extratrees  0.8305697  0.6287598
##   5      7             gini        0.8462879  0.6667411
##   5      7             extratrees  0.8339280  0.6365362
##   5     10             gini        0.8429296  0.6588122
##   5     10             extratrees  0.8305697  0.6286259
##   6      1             gini        0.8440407  0.6627376
##   6      1             extratrees  0.8316806  0.6313039
##   6      3             gini        0.8372988  0.6485325
##   6      3             extratrees  0.8339533  0.6362663
##   6      5             gini        0.8406696  0.6552537
##   6      5             extratrees  0.8317058  0.6319900
##   6      7             gini        0.8462876  0.6668471
##   6      7             extratrees  0.8317058  0.6319218
```

```
##   6     10             gini        0.8429296  0.6600389
##   6     10             extratrees  0.8350769  0.6387102
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were mtry = 5, splitrule = gini
##  and min.node.size = 7.
```

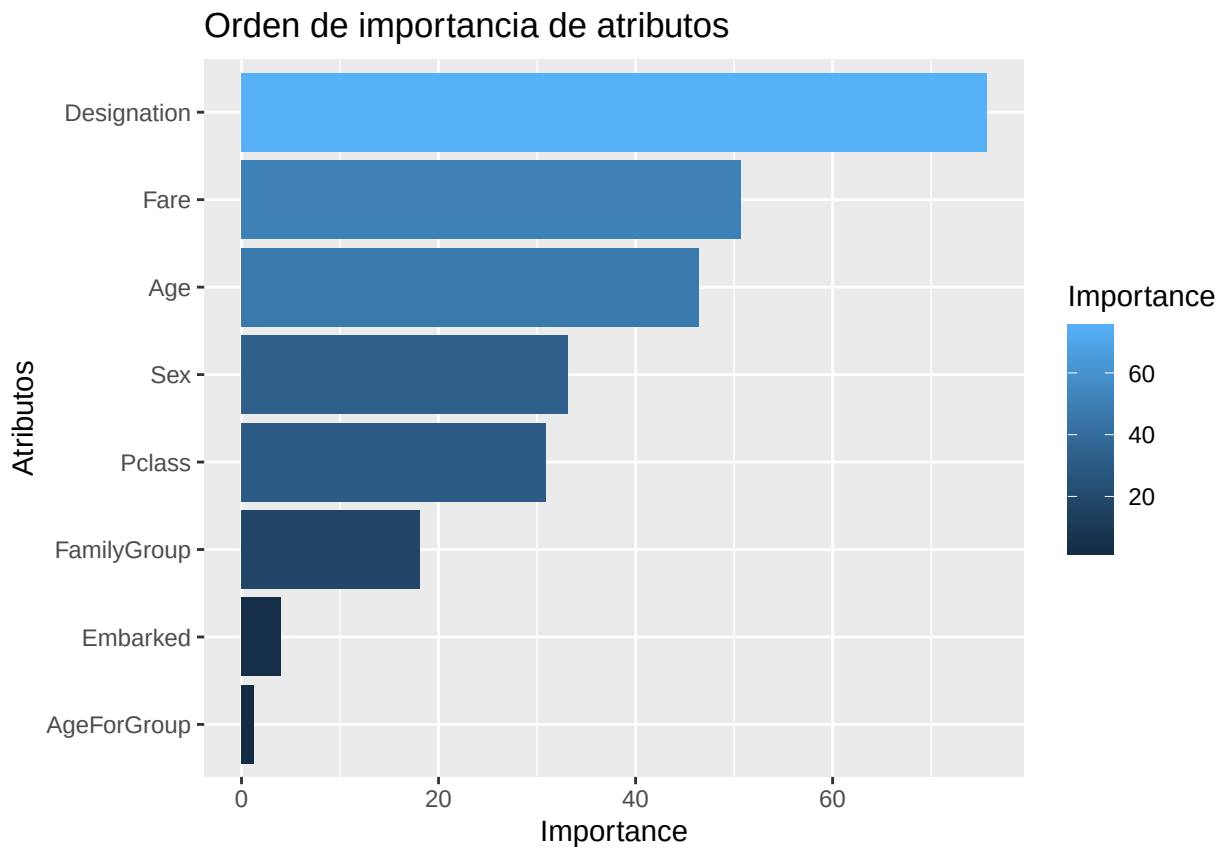
Se procede a minimizar la impureza de los datos con función ranger().

```
library(ranger)
```

```
rf_impurity <- ranger(
  formula = Survived ~ .,
  data = titanic_rf,
  num.trees = 2000,
  mtry = model_rf$bestTune$mtry,
  min.node.size = model_rf$bestTune$min.node.size,
  sample.fraction = .80,
  replace = FALSE,
  importance = "impurity",
  respect.unordered.factors = "order",
  verbose = FALSE,
  seed = 123
)
```

Mediante el análisis de importancia de componentes se evidencia un papel relevante en Fare, Age y Sex.

```
vip1 <- vip(rf_impurity, bar="FALSE")

vip1$data %>%
  ggplot(aes(x = Importance, y = reorder(Variable, Importance), fill = Importance)) +
    geom_col() +
    labs(
      title = "Orden de importancia de atributos",
      y = "Atributos"
    )
```



La precisión de este modelo es 90% y al crear la matriz de confusión obtenemos alta especificidad de 96% concluyendo que es modelo es muy preciso.

El 89% de todas las muertes predichas son correctas. El 93% de sobrevivientes predichas son correctas.

```
titanic_rf$survival_pred <- predict(model_rf, titanic_rf)
model_prob <- predict(model_pls, newdata = titanic_rf, type = "prob")

titanic_rf <- titanic_rf %>%
  mutate(survival_prob = model_prob[,2])

cfMatrix <- confusionMatrix(
  data = relevel(titanic_rf$survival_pred, ref = "1"),
  reference = relevel(titanic_rf$Survived, ref = "1")
)

cfMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   0
##          1 276  24
##          0  66 525
##
##                Accuracy : 0.899
##                  95% CI : (0.8773, 0.918)
##     No Information Rate : 0.6162
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7814
##
##  Mcnemar's Test P-Value : 1.548e-05
##
##             Sensitivity : 0.8070
##             Specificity : 0.9563
##          Pos Pred Value : 0.9200
##          Neg Pred Value : 0.8883
##              Prevalence : 0.3838
##          Detection Rate : 0.3098
##    Detection Prevalence : 0.3367
##       Balanced Accuracy : 0.8817
##
##        'Positive' Class : 1
##
```

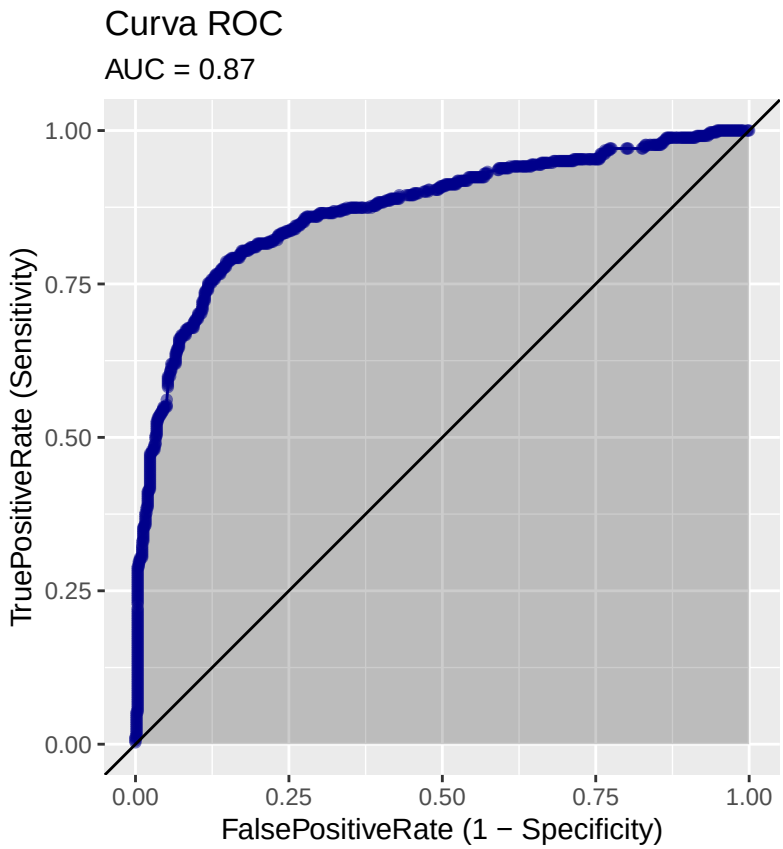
Dibujamos la curva ROC.En el cuadro se observa el resultado de 0.87, es decir, la tasa de verdaderos positivos está por encima de la diagonal.

```
ROC <- roc(titanic_rf$Survived, as.numeric(titanic_rf$survival_pred))
```

```
## Setting levels: control = 0, case = 1
```

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

```
titanic_train2 <- titanic_rf %>%
  mutate(Survived = as.numeric(as.character(Survived)),
         survival_prob = as.numeric(as.character(survival_prob)))

ROCPlot(titanic_train2, "survival_prob", "Survived",
        truthTarget = TRUE, title = "Curva ROC") +
  labs(
    title = "Curva ROC"
  )
```



Ahora estimamos la supervivencia del conjunto de prueba con el modelo creado.

```
titanic_train_recipe <- new.titanic.train %>%
  mutate(Survived = ifelse(Survived =="1",0,1)) %>%
  select(-c("Name", "Ticket", "Cabin","SibSp", "Parch"))

test_recipe <- recipe(Survived ~ ., data = titanic_train_recipe)

prepare <- prep(test_recipe, training = titanic_train_recipe, strings_as_factors = TRUE)

titanic_rf_test <- new.titanic.test %>%
  mutate(Survived = as.integer(0)) %>%
  select(c("PassengerId","Survived", "Pclass","Sex","Age","AgeForGroup","Fare","Embarked",
 "Designation","FamilyGroup"))

baked_rf_test <- bake(prepare, new_data = titanic_rf_test)

baked_rf_test <- baked_rf_test %>%
  mutate(Pclass = factor(Pclass),
         Sex = factor(Sex))

baked_rf_test$prediction_rf <- predict(model_rf, baked_rf_test)

baked_rf_test %>%  select(PassengerId,prediction_rf)
```

```
## # A tibble: 418 x 2
##    PassengerId prediction_rf
```

```
##          <int> <fct>
##  1         892 0
##  2         893 0
##  3         894 0
##  4         895 0
##  5         896 1
##  6         897 0
##  7         898 0
##  8         899 0
##  9         900 1
## 10         901 0
## # ... with 408 more rows
```

El resultado en el conjunto test aplicando el modelo de regresión es que de 418 pasajeros, 148 personas sobrevivieron.

```
summary(baked_rf_test$prediction_rf)
```

```
##   0   1
## 270 148
```

5.3 Modelo Árbol de decisión

Este modelo predice la variable de respuesta con un conjunto de reglas binarias repartiendo las observaciones en función de sus atributos. Para ajustar la profundidad máxima del árbol, estableceremos validación cruzada con el método rpart2.

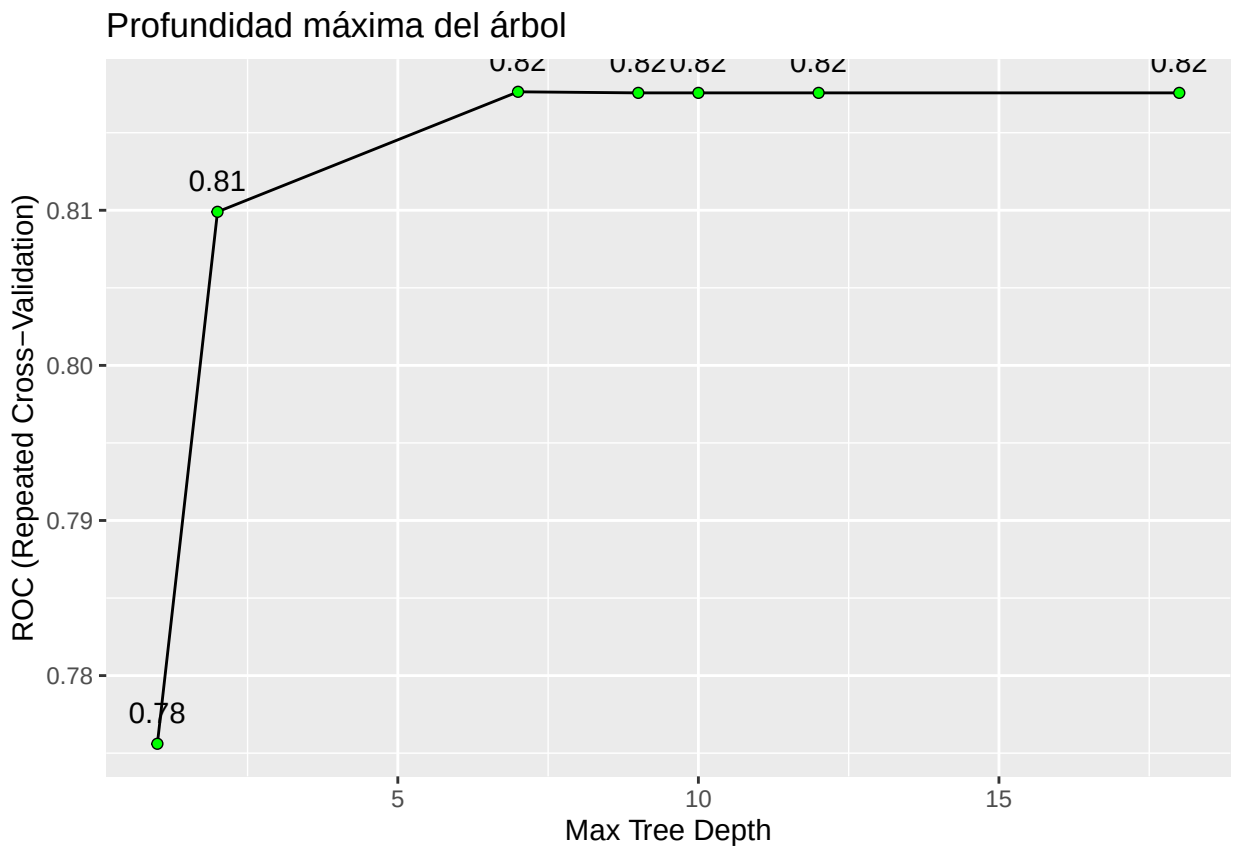
```
titanic_tree <- new.titanic.train[,c("Survived","Pclass","Sex","Age","AgeForGroup","Fare",
 "Embarked","Designation","FamilyGroup")]

set.seed(123)

ctrl <- trainControl(
                    method = "repeatedcv",
                    number = 10,
                    repeats = 3,
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE
                    )

model_tree <- train(
  make.names(Survived) ~ .,
  data = titanic_tree,
  method = "rpart2",
  trControl = ctrl,
  tuneLength = 10,
  metric = "ROC"
)
```

```
## note: only 7 possible values of the max tree depth from the initial fit.
##  Truncating the grid to 7 .
```

Graficamos la profundidad máxima del árbol obteniendo un valor de 7 con ROC = 0.82

```
ggplot(model_tree) +
  geom_point(color = "green", size = 1) +
  geom_text(aes(x = maxdepth, y = ROC, label = round(ROC,2)),
```

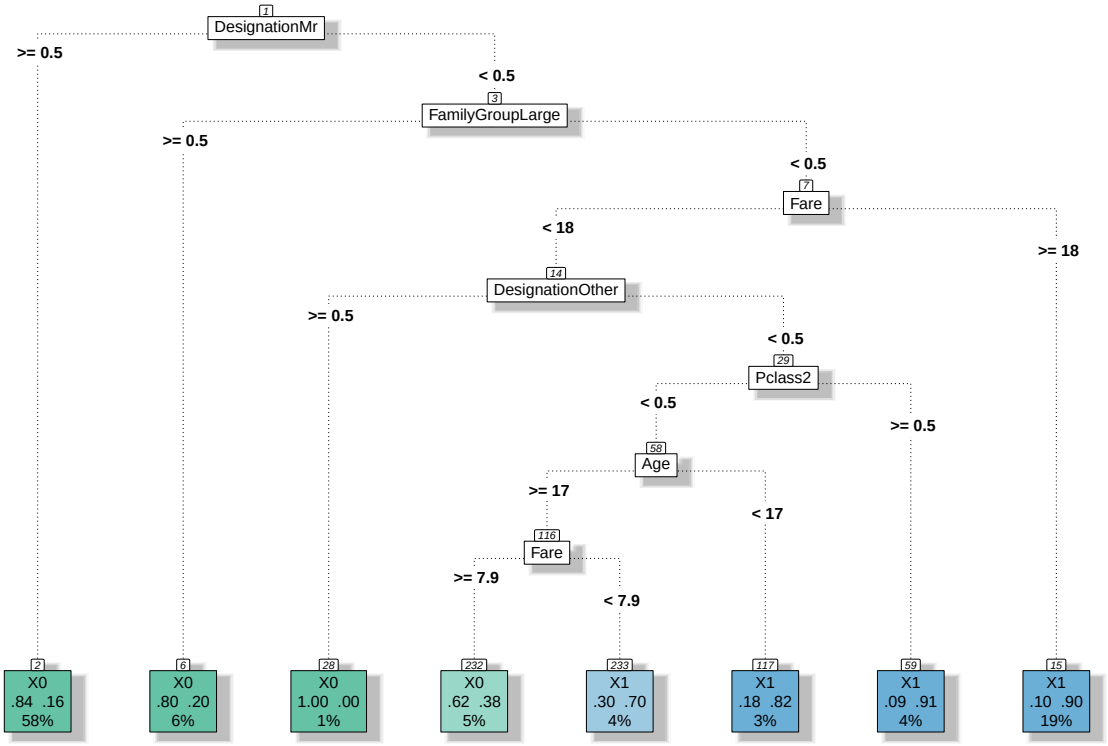
```
          vjust = -1) +
  labs(
    title = "Profundidad máxima del árbol"
  )
```



Precisamos las caracterísitcas más importantes de supervivencia dibujando el árbol de decisiones.

```
library(rattle)
```

```
fancyRpartPlot(model_tree$finalModel,type = 5,  palettes = "BuGn",pal.thresh = 0.5,
 cex = 0.5,round = 0, pal.node.fun = TRUE)
```



Las variables que se destacan en el árbol son: Fare, Pclass (2), y Age.

Al revisar los resultados del modelo notamos que la precisión del modelo es 83.73% y una especificidad de 93.26% significa que el modelo es muy preciso en la predicción.

El 82.58% de todas las muertes predichas son correctas. El 86.35% de sobrevivientes predichas son correctas.

```
titanic_tree$survival_pred <- predict(model_tree, titanic_tree)
model_prob <- predict(model_tree, newdata = titanic_tree, type = "prob")

titanic_tree <- titanic_tree %>%
  mutate(survival_prob = model_prob[,2]) %>%
  mutate(survival_pred = ifelse(survival_pred == "X0", 0, 1)) %>%
  mutate(survival_pred = factor(survival_pred))


cfMatrix <- confusionMatrix(
  data = relevel(titanic_tree$survival_pred, ref = "1"),
  reference = relevel(titanic_tree$Survived, ref = "1")
)

cfMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   0
##          1 234  37
```

```
##          0 108 512
##
##                Accuracy : 0.8373
##                  95% CI : (0.8114, 0.8609)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6419
##
##  Mcnemar's Test P-Value : 6.13e-09
##
##             Sensitivity : 0.6842
##             Specificity : 0.9326
##          Pos Pred Value : 0.8635
##          Neg Pred Value : 0.8258
##              Prevalence : 0.3838
##          Detection Rate : 0.2626
##    Detection Prevalence : 0.3042
##       Balanced Accuracy : 0.8084
##
##        'Positive' Class : 1
##
```

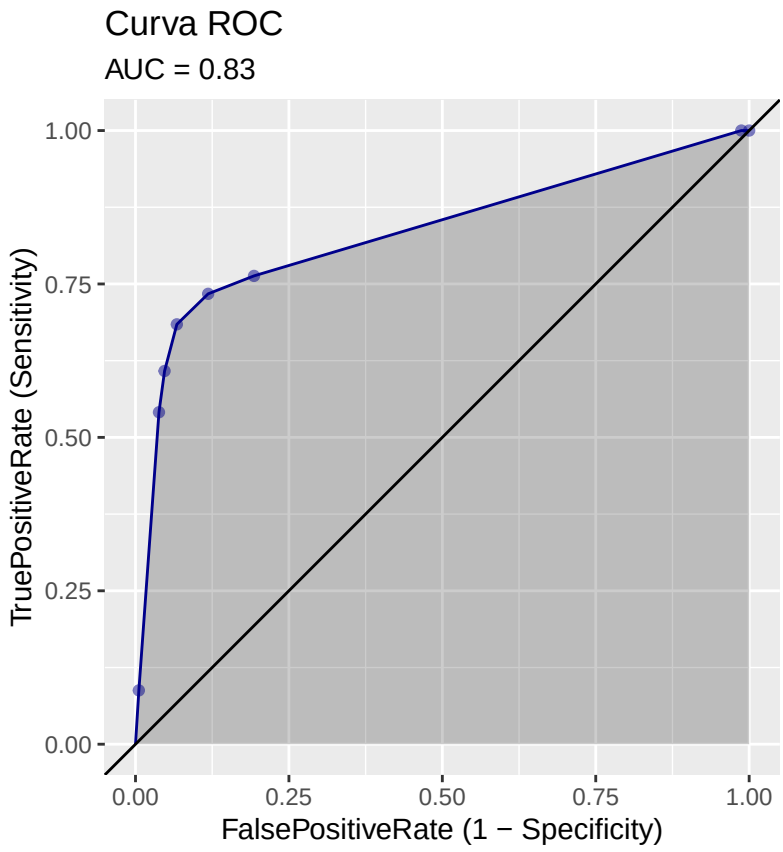
En la curva ROC se evidencia un área bajo la curva de 0.83

```
ROC <- roc(titanic_tree$Survived, as.numeric(titanic_tree$survival_pred))
```

```
## Setting levels: control = 0, case = 1
```

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

```
titanic_train2 <- titanic_tree %>%
  mutate(Survived = as.numeric(as.character(Survived)),
         survival_prob = as.numeric(as.character(survival_prob)))

ROCPlot(titanic_train2, "survival_prob", "Survived",
        truthTarget = TRUE, title = "Curva ROC") +
  labs(
    title = "Curva ROC"
  )
```



Procedemos a estimar la supervivencia con el modelo creado.

```
titanic_train_recipe <- new.titanic.train %>%
  mutate(Survived = ifelse(Survived =="1",0,1)) %>%
  select(-c("Name", "Ticket", "Cabin","SibSp", "Parch"))

test_recipe <- recipe(Survived ~ ., data = titanic_train_recipe)

prepare <- prep(test_recipe, training = titanic_train_recipe, strings_as_factors = TRUE)

titanic_tree_test <- new.titanic.test %>%
  mutate(Survived = as.integer(0)) %>%
  select(c("PassengerId","Survived", "Pclass","Sex","Age","AgeForGroup","Fare","Embarked",
           "Designation","FamilyGroup"))

baked_tree_test <- bake(prepare, new_data = titanic_tree_test)

baked_tree_test <- baked_tree_test %>%
  mutate(Pclass = factor(Pclass),
         Sex = factor(Sex))

baked_tree_test$prediction_tree <- predict(model_tree, baked_tree_test)

baked_tree_test <- baked_tree_test %>%
  mutate(tree_prediction = ifelse(prediction_tree == "X0", 0, 1)) %>%
  mutate(tree_prediction = factor(prediction_tree))
```

```
baked_tree_test %>%  select(PassengerId,prediction_tree)
```

```
## # A tibble: 418 x 2
##    PassengerId prediction_tree
##          <int> <fct>
##  1         892 X0
##  2         893 X1
##  3         894 X0
##  4         895 X0
##  5         896 X0
##  6         897 X0
##  7         898 X1
##  8         899 X0
##  9         900 X1
## 10         901 X0
## # ... with 408 more rows
```

El resultado en el conjunto test aplicando el modelo de regresión es que de 418 pasajeros, 138 personas sobrevivieron.

```
summary(baked_tree_test$prediction_tree)
```

```
##  X0  X1
## 280 138
```

5.4 Comparación de modelos

La siguiente tabla muestra los resultados de los tres modelos utilizados para probar la predicción de supervivencia de pasajeros del Titanic.

En general los tres modelos tuvieron una precisión muy buena. El modelo de Regresión Logística estima exactitud de 82.60%. Con Random Forest la precisión incrementa a un 90%. Así como, aplicando el Árbol de decisión tiene la precisión del 83.73%.

La tasa de éxito más alta entre muertes y supervivencia la determina la curva de ROC que compara la tasa de verdaderos positivos con falsos positivos. El mayor valor de ROC (0.87) se presentó en los modelos de Regresión Logística y Random Forest.

Modelo	Precisión	Predicción de muerte	Predicción de supervivencia
1	82.60%	255 61.00%	163 39.00%
2	90.00%	270 64.59%	148 35.41%
3	83.73%	280 66.99%	138 33.01%

5.5 Resultados

La precisión del Modelo 2 (Random Forest) es la más alta entre todos los demás modelos anteriores. Por lo tanto, seleccionaremos las predicciones del Modelo 2 para gráfico y tablas de análisis de resultados.

```
titanic.test.surv <- new.titanic.test
titanic.test.surv$Survived <- baked_rf_test$prediction_rf
```

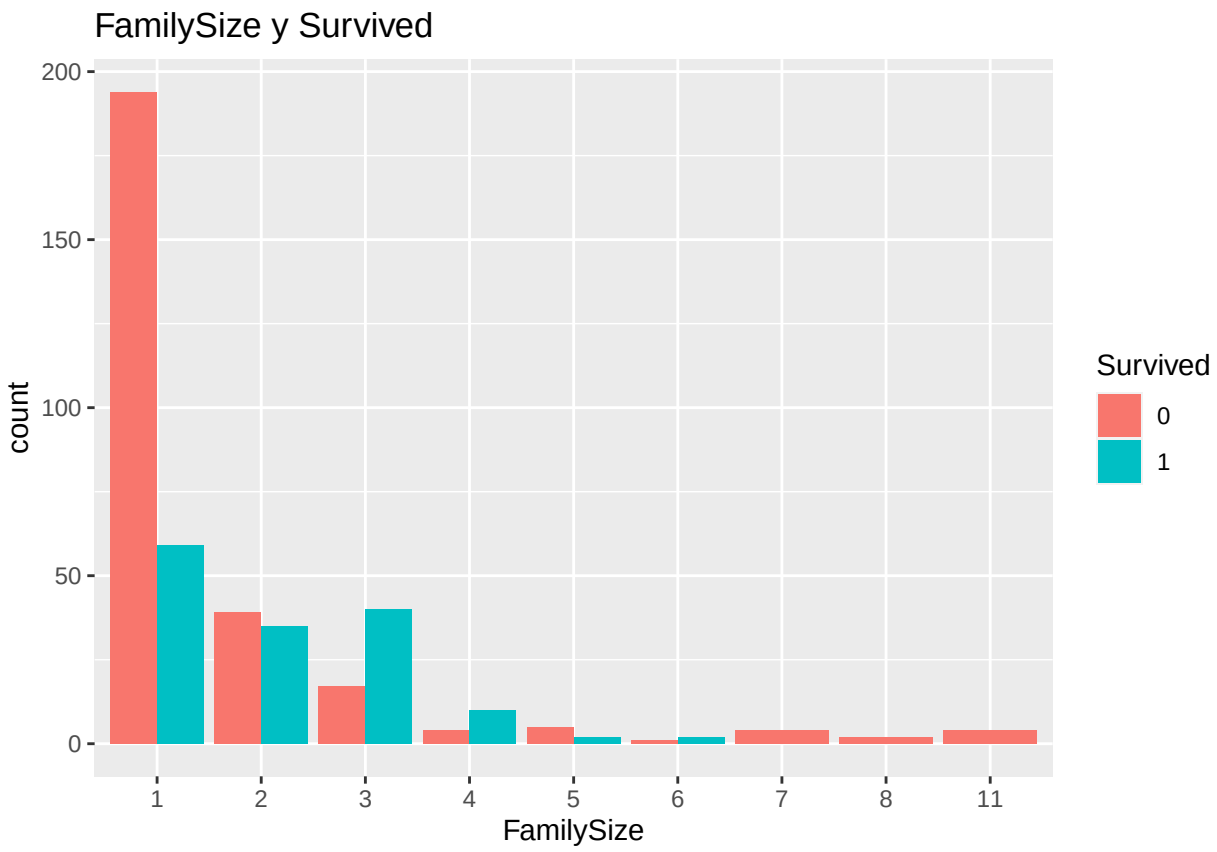
Se realiza análisis exploratorio de los datos calculados en el conjunto de datos de prueba en función de nuestra variable resultado "Survived".

```
# Cargar la librería ggplo2
#library(ggplot2)
```

```
#library(dplyr)
#library(grid)
#library(gridExtra)
```

Mostramos la relación entre el tamaño de la familia y supervivencia mediante un diagrama de barras.

```
# Mostrar número de casos FamilySize y Survived
ggplot(titanic.test.surv, aes(x = FamilySize, fill = Survived), na.rm = TRUE)+
  geom_bar(stat = "count", position = "dodge")+
  ggtitle("FamilySize y Survived")
```




```
# Tabla de frecuencia de tamaño de familia y supervivencia
prop.table(table(titanic.test.surv$FamilySize, titanic.test.surv$Survived),
 margin = 1) %>% round(digits = 3)
```

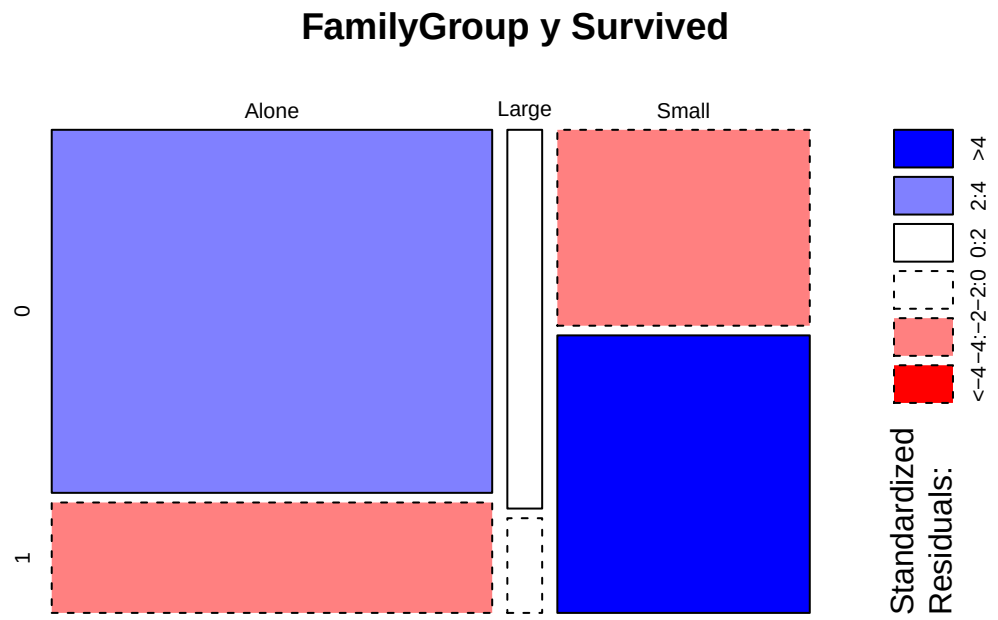
```
##
##          0     1
##   1  0.767 0.233
##   2  0.527 0.473
##   3  0.298 0.702
##   4  0.286 0.714
##   5  0.714 0.286
##   6  0.333 0.667
##   7  1.000 0.000
##   8  1.000 0.000
##   11 1.000 0.000
```

El gráfico resume las siguientes características respecto a familia:

- La muerte predomina en pasajeros que abordaron solos o con muchos familiares.
- La tasa de supervivencia es más alta para un solo pasajero.

Además se puede utilizar el grupo de la familia como: solo, familia pequeña y familia grande. Para esto utilizamos un gráfico basado en el grupo de la familia.

```
mosaicplot(table(titanic.test.surv$FamilyGroup, titanic.test.surv$Survived),
  main = "FamilyGroup y Survived", shade = TRUE)
```




```
# Tabla de frecuencia de grupo de familia y supervivencia
prop.table(table(titanic.test.surv$FamilyGroup, titanic.test.surv$Survived),
 margin = 1) %>% round(digits = 3)
```

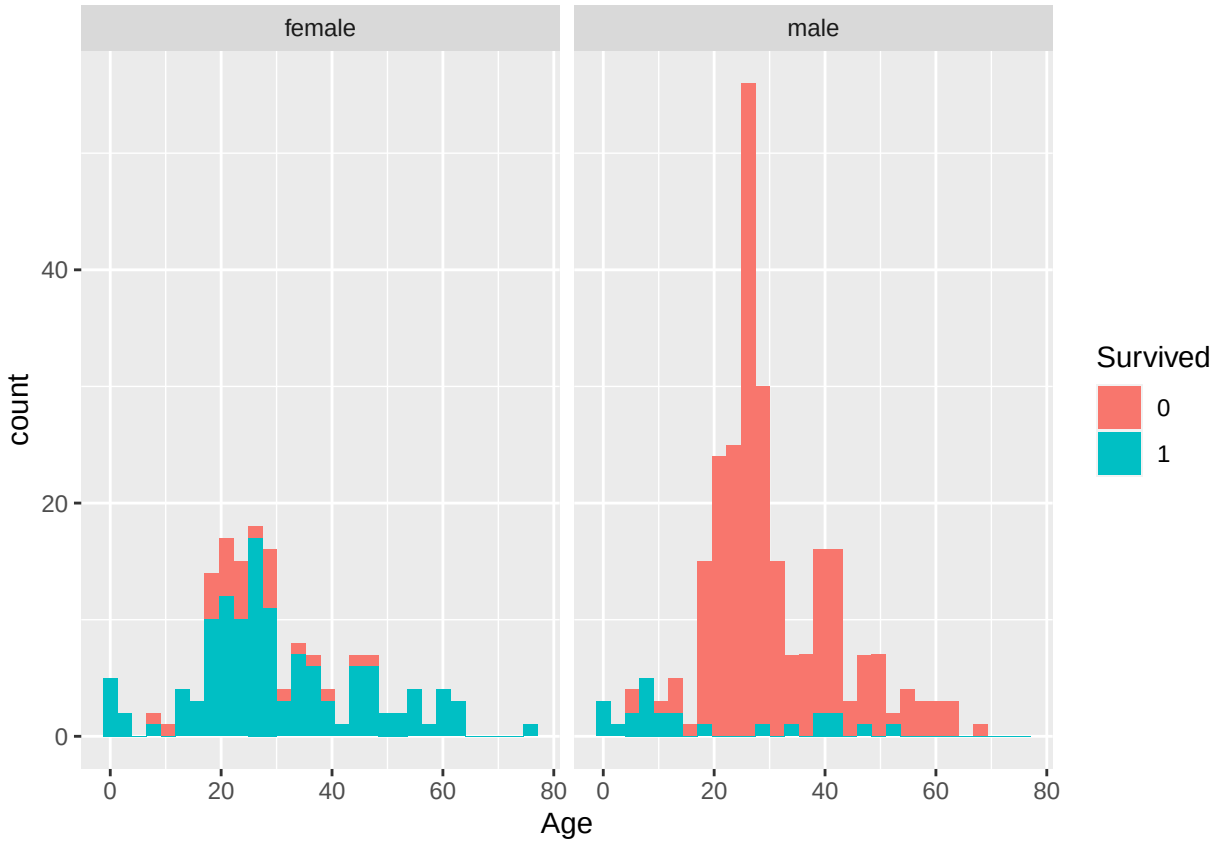
```
##
##              0     1
##   Alone 0.767 0.233
##   Large 0.800 0.200
##   Small 0.414 0.586
```

La distribución muestra que las familias pequeñas tienen tasa de supervivencia más alta que las familias grandes.

A continuación representamos la relación de edades de los pasajeros y supervivencia por cada sexo a través de un histograma.

```
# Mostrar número de casos Age, Sex y Survived
ggplot(titanic.test.surv, aes(x = Age, fill = Survived))+
  geom_histogram()+
  facet_grid(.~Sex)
```

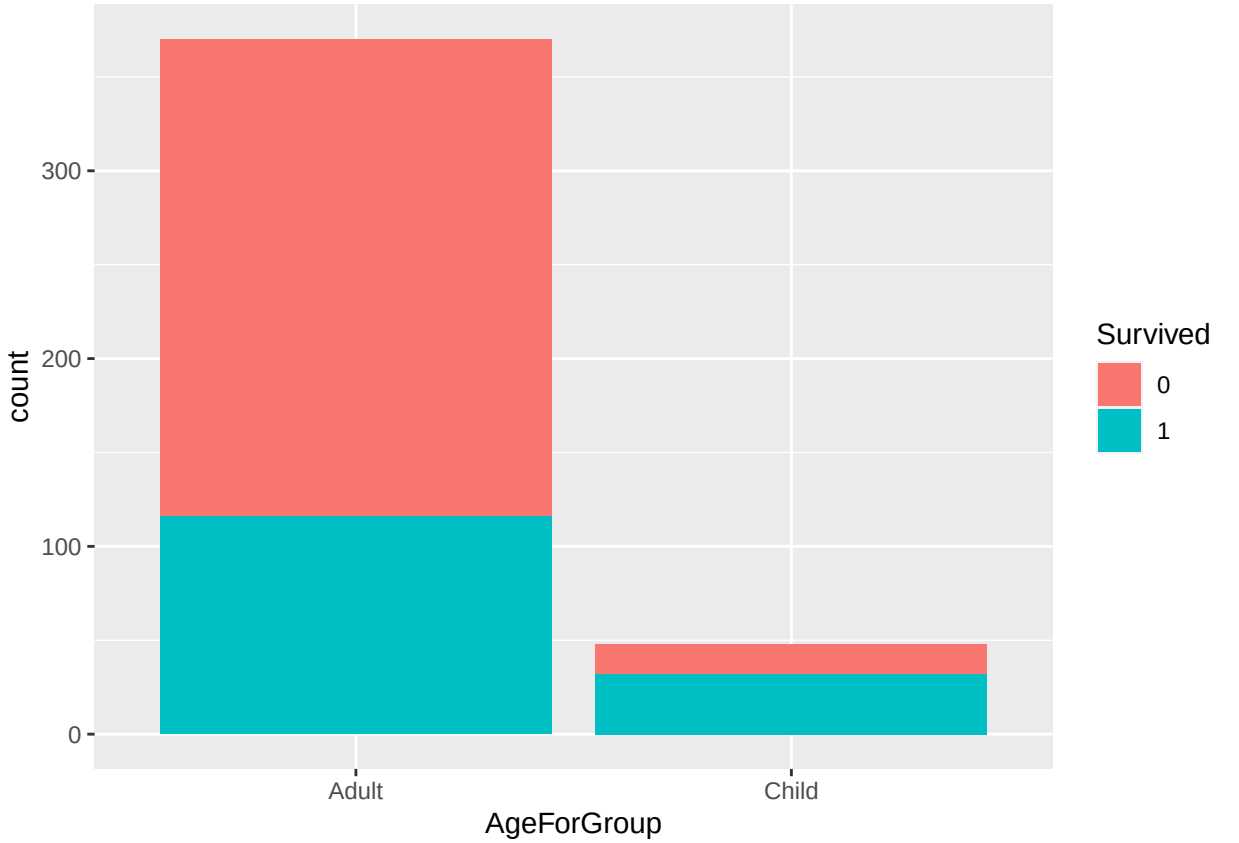
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- En el gráfico anterior existe un patrón por edades similar tanto para sobrevivientes y no sobrevivientes.
- El nivel de supervivencia de mujeres es mayor al de hombres ya que se les da prioridad.

Considerando el agrupamiento de menores y mayores de edad se examina con barras la supervivencia de esta clasificación.

```
# Mostrar número de casos FamilySize y Survived
ggplot(titanic.test.surv, aes(x = AgeForGroup, fill = Survived))+
  geom_bar(stat = "count")
```




```
# Tabla de frecuencia de grupo de edad y supervivencia
prop.table(table(titanic.test.surv$AgeForGroup, titanic.test.surv$Survived),
 margin = 1) %>% round(digits = 3)
```

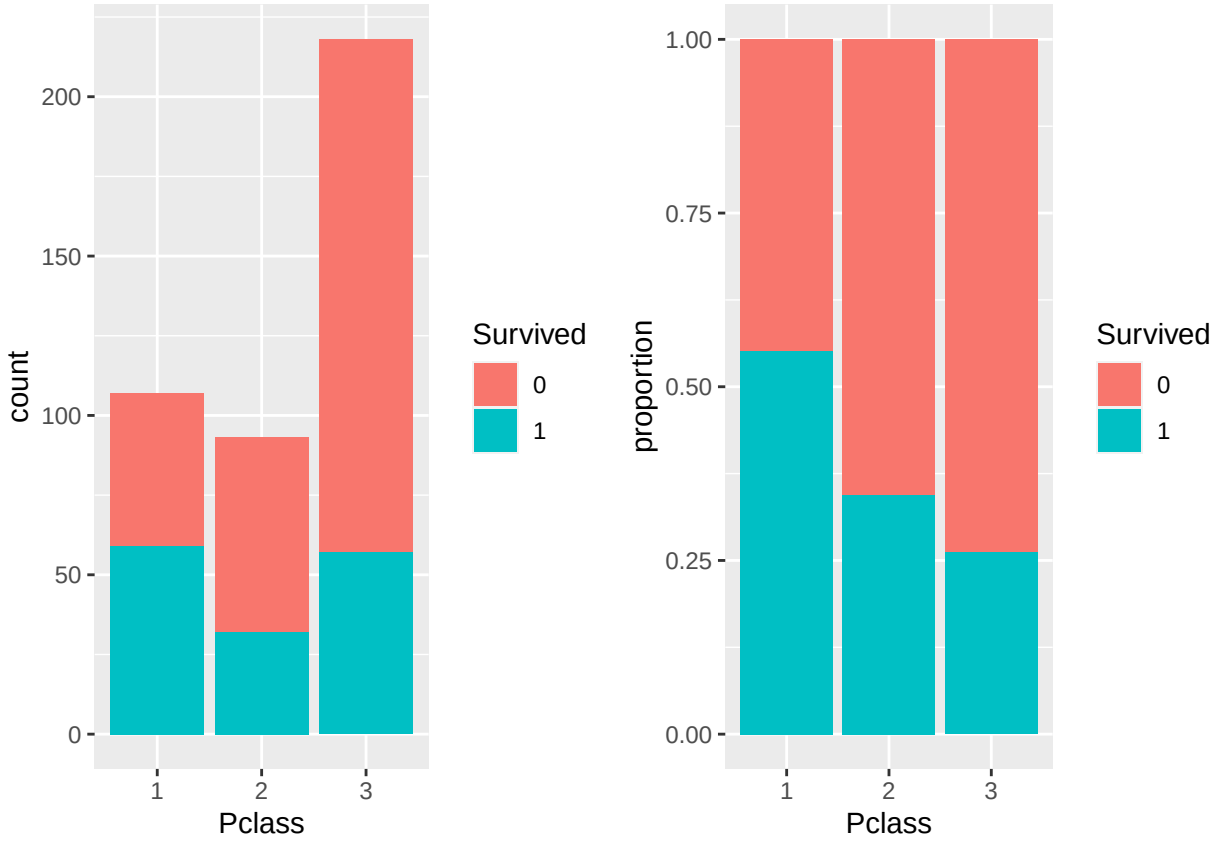
```
##
##              0     1
##   Adult 0.686 0.314
##   Child 0.333 0.667
```

El resultado de la tabla anterior indica lo siguiente:

- Más de un 60% de menores de edad abordo sobreviven, esto confirma que por norma la tripulación les da prioridad en botes salvavidas.
- Mientras que los mayores de edad tienen menos supervivencia de alrededor del 31%, porque algunos ceden su espacio a los menores.

Ahora se realiza un diagrama de la supervivencia según la clase social de pasajeros.

```
# Mostrar número de casos Pclass y Survived
pclass.surv.plot <- ggplot(titanic.test.surv, aes(x = Pclass, fill = Survived))+
  geom_bar(stat = "count")

# Mostrar proporción de Pclass y Survived
pclass.surv.prop.plot <- ggplot(titanic.test.surv, aes(x = Pclass, fill = Survived))+
  geom_bar(position = "fill")+
  labs(y = "proportion")

grid.arrange(pclass.surv.plot, pclass.surv.prop.plot, ncol = 2)
```



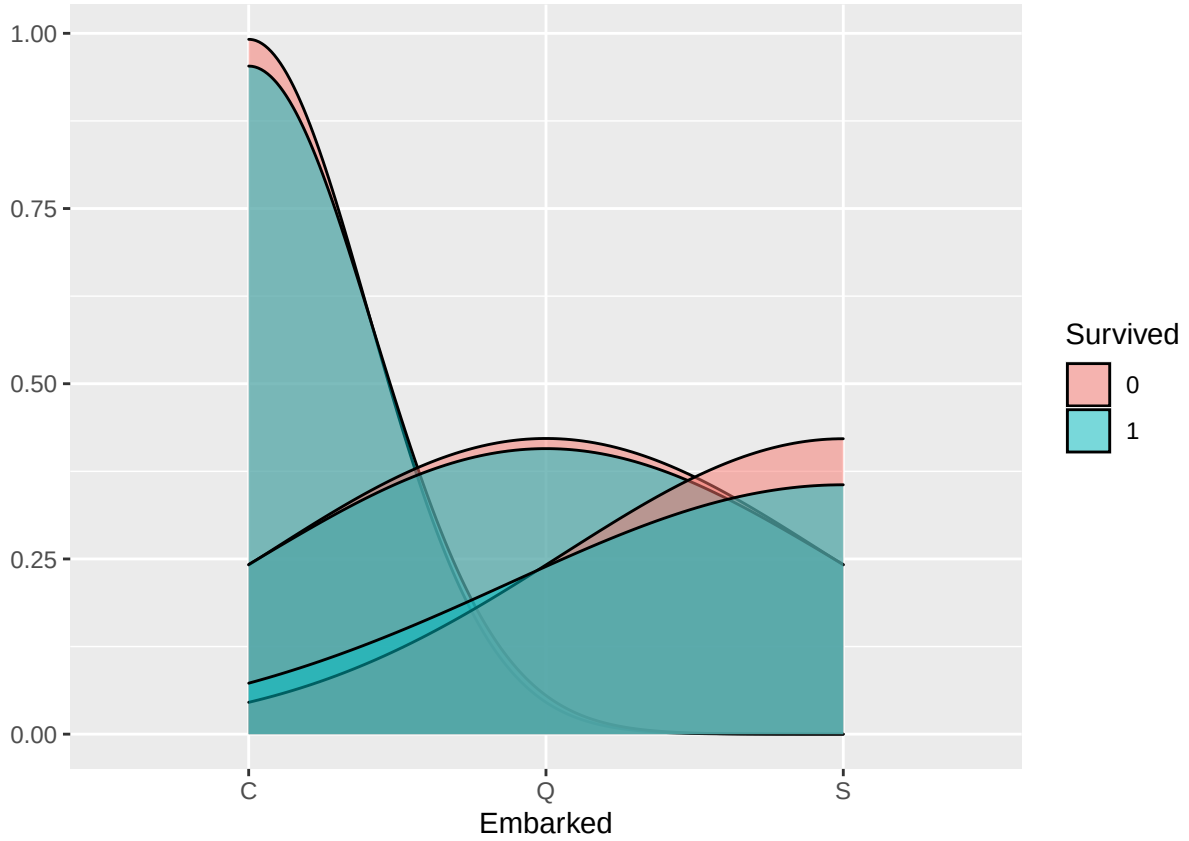

```
# Tabla de frecuencia de clase social y supervivencia
prop.table(table(titanic.test.surv$Pclass, titanic.test.surv$Survived),
 margin = 1) %>% round(digits = 3)
```

```
##
##         0     1
##   1 0.449 0.551
##   2 0.656 0.344
##   3 0.739 0.261
```

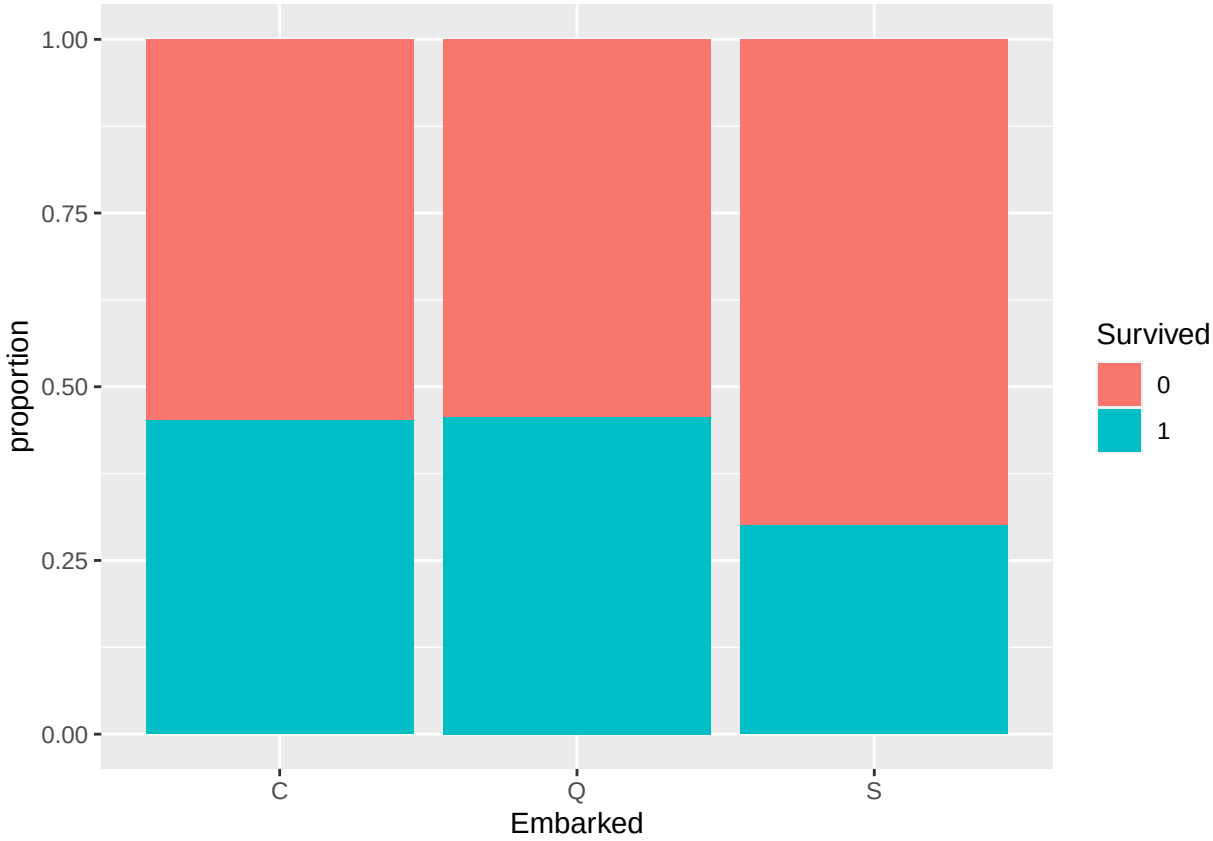
El gráfico muestra la diferencia que hace la clase social de la persona respecto a su supervivencia:

- Se puede citar que más del 50% de pasajeros de clase 1 sobrevive en caso que la tripulación embarque en los botes salvavidas a las clases superiores cuando no quede una mujer o niño.
- Menos de un tercio de los pasajeros de tercera clase sobrevive cuando su cubierta no tiene botes salvavidas y hay menos probabilidad de encontrar el camino a otras cubiertas dentro del barco.

```
par(mfrow = c(1, 2))
# Mostrar densidad Embarked y Survived
qplot(Embarked, data = titanic.test.surv, geom = "density", fill = Survived, alpha = I(0.5))
```




```
# Mostrar proporción de Embarked y Survived
ggplot(titanic.test.surv, aes(x = Embarked, fill = Survived))+
  geom_bar(position = "fill")+
  labs(y = "proportion")
```



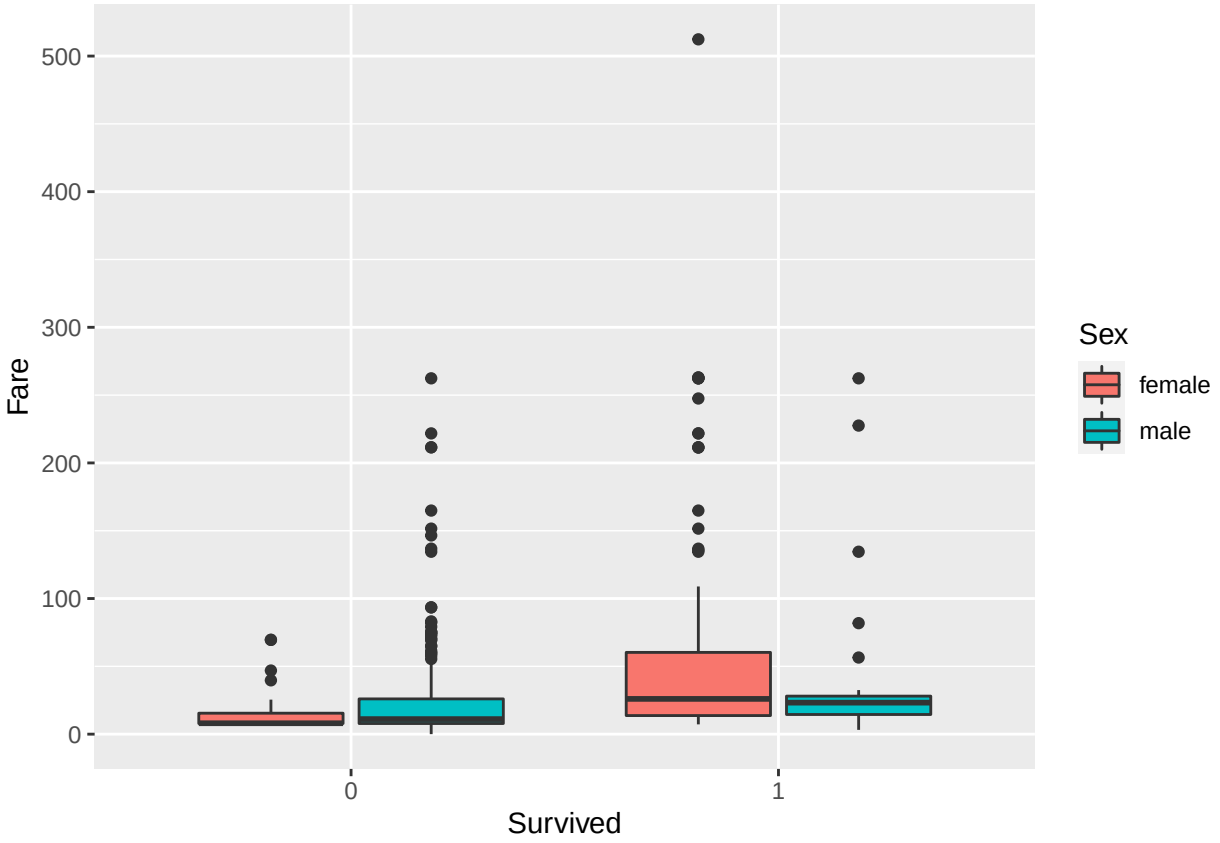

```
# Tabla de frecuencia de puerto y supervivencia
prop.table(table(titanic.test.surv$Embarked, titanic.test.surv$Survived),
 margin = 1) %>% round(digits = 3)
```

```
##
##         0     1
##   C 0.549 0.451
##   Q 0.543 0.457
##   S 0.700 0.300
```

- La densidad muestra que el puerto Queenstown "Q" es el puerto donde la tasa de supervivencia es más alta que el puerto Cherburgo "C" y Southampton "S", donde la tasa de no supervivencia es mayor que la supervivencia.
- Alrededor del 45.7% de los pasajeros embarcados en el puerto Queenstown "Q" lograron sobrevivir en relación del 45.1% y el 30% de los pasajeros embarcados en el puerto Cherburgo "C" y Southampton "S" respectivamente.

Ahora se analiza la tarifa promedio y más alta pagada en relación al sexo y supervivencia.

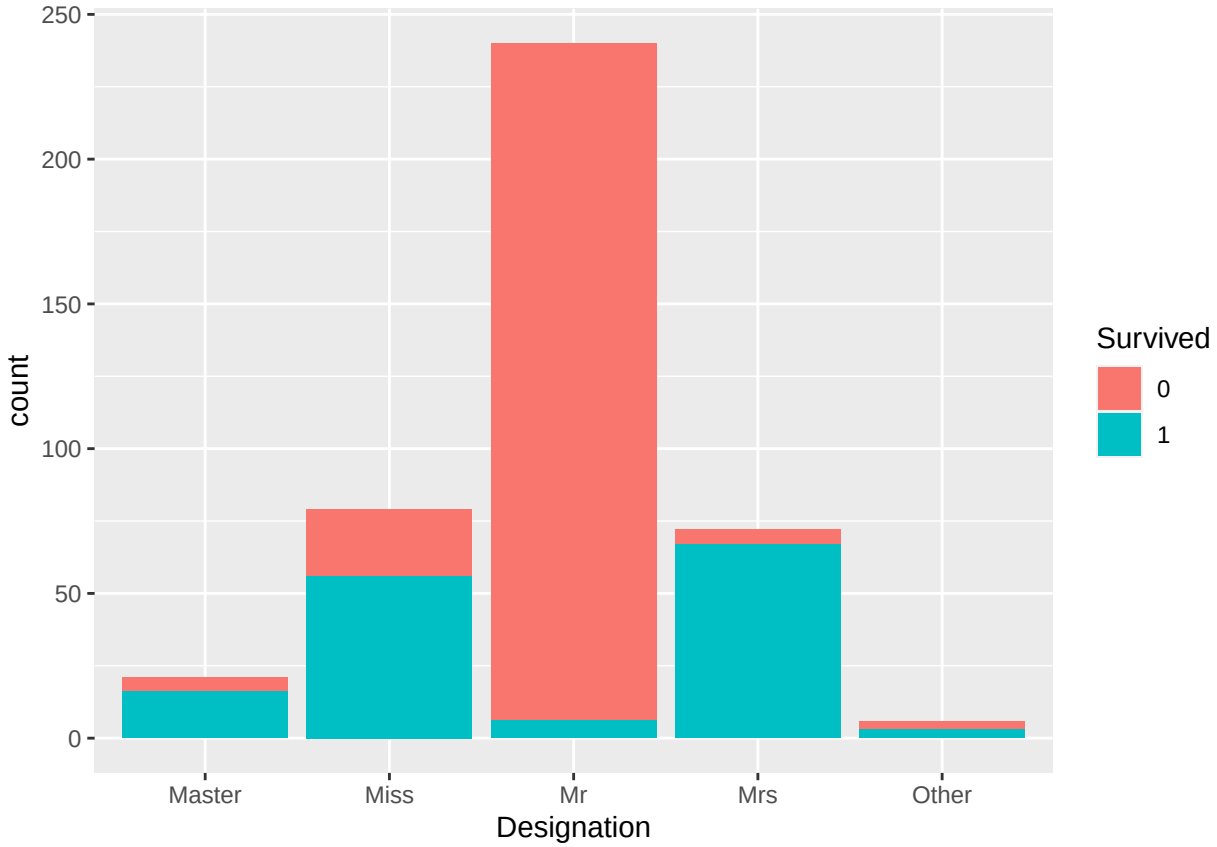
```
ggplot(titanic.test.surv, aes(x = Survived, y = Fare, fill = Sex))+
  geom_boxplot()
```



- En el gráfico se puede deducir que la tarifa promedio pagada por las mujeres es más alta que la de los hombres.
- La tarifa promedio menor se encuentra en el grupo de no sobreviventes.

Se analiza alguna relación del título con la supervivencia.

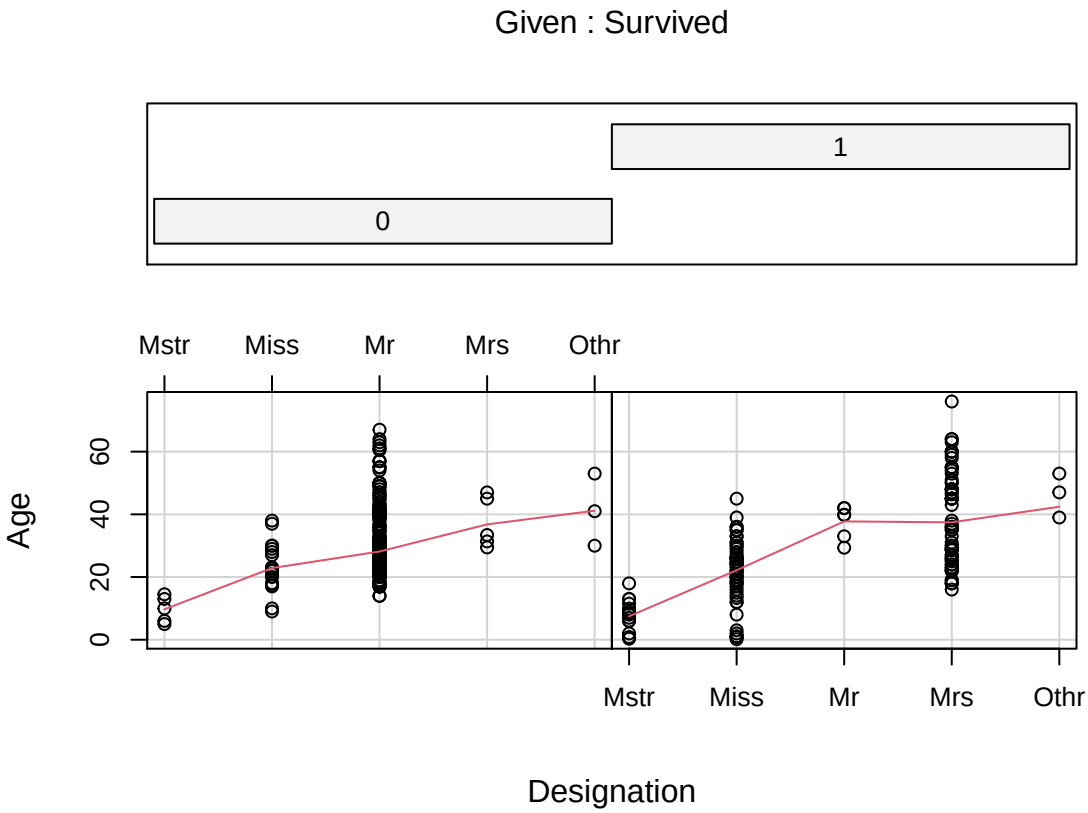
```
# Mostrar número de casos Designation y Survived
ggplot(titanic.test.surv, aes(x = Designation, fill = Survived))+
  geom_bar(stat = "count")
```




```
# Tabla de frecuencia de título y supervivencia
prop.table(table(titanic.test.surv$Designation, titanic.test.surv$Survived),
 margin = 1) %>% round(digits = 3)
```

```
##
##                0     1
##   Master  0.238 0.762
##   Miss    0.291 0.709
##   Mr      0.975 0.025
##   Mrs     0.069 0.931
##   Other   0.500 0.500
```

```
coplot(Age~Designation|Survived, data = titanic.test.surv, panel = panel.smooth)
```



- El gráfico muestra que las personas desginadas como: Señora, Señorita y Máster tienen alta tasa de supervivencia.
- En cambio el título de Señor tiene la más alta tasa de no supervivencia en todas las barras.
- No se observa ninguna diferencia para las edades promedio de los sobrevivientes y no sobrevivientes en el título designado.

Finalmente podemos usar la predicción de los datos de prueba originales y almacenar el resultado en un archivo CSV.

```
titanic.test.csv <- titanic.test.surv[, c("PassengerId", "Survived", "Pclass",
 "Name", "Sex", "Age", "SibSp", "Parch", "Ticket", "Fare", "Cabin", "Embarked")]
write.csv2(titanic.test.csv, file = "prediction.csv", row.names = FALSE, quote = FALSE)
```

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

- El atributo "PassengerId" es un código único que permitiría una integración vertical de los conjuntos de entrenamiento y prueba descargados desde kaggle. Sin embargo, no es adecuado fusionarlos porque el conjunto de pruebas se empleará para estimar la supervivencia una vez generado los modelos de predicción con el conjunto de datos de entrenamiento.
- Como estrategia a nuestro análisis se emplearon nuevos atributos de agrupamiento, así por ejemplo: el atributo Designation que contiene la designación honorífica de las personas; FamilySize contiene el

grupo de familia según su número y AgeForGroup que contiene la clasificación de las personas según su edad.

- Los valores ausentes se muestran en cuatro variables: Cabin, Age, Fare y Embarked siendo significativa en cantidad para Cabin y Age en ambos conjuntos de datos. La variable Cabin no es necesaria tratarla y se descarta para el análisis. Por el contrario, para la variable Age aplicamos el método de predicción missForest. Con las variables Fare y Embarked reemplazamos los valores perdidos por la medida de tendencia central respectiva.
- En el caso de valores extremos se detectó outliers en los atributos Age y Fare. Ambos, luego de ser analizados se consideran valores posibles.
- Con la creación de 3 modelos: Regresión Logística, Random Forest y Árbol de decisiones; se identificaron los factores más influyentes en la supervivencia de un pasajero y son: Sex, Pclass, Fare y Age. El mejor modelo predictivo es Random Forest con 90% de precisión.
- En relación a las preguntas planteadas para nuestro proyecto y considerando el mejor modelo predictivio, podemos concluir que:
- Las mujeres tienen más probabilidades de supervivencia que los hombres.
- La tasa de supervivencia es más alta para un solo pasajero.
- Más del 50% de pasajeros de primera clase sobrevive, en tanto que, menos de un tercio de los pasajeros de tercera clase sobrevive.
- El puerto Q=Queenstown tiene una tasa mayor de supervivencia que los demás puertos, pero siendo la diferencia no muy notoria y obedeciendo al modelo de predicción, la variable Embarked no es un factor influyente.
- Considerando la clasificación de las personas según su edad, notamos que entre niños y adultos, los adultos representan un gran número en fallecidos y sobrevivientes si se compara con los niños.

7 Recursos

- Salazar, C. (2020). Limpieza de datos. RPubs by RStudio. https://rpubs.com/camilamila/limpieza
- (2020). Titanic Data Transformation. RPubs by RStudio. https://rstudio-pubs-static.s3.amazonaws.com/421800_30e830cbb8414b6ea8854dd0be118d22.html
- Donges, N. (2018). Predicting the Survival of Titanic Passengers. Towards Data Science. https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8
- Hammer, B. (2017). Titanic - Machine Learning from Disaster. Kaggle. https://www.kaggle.com/c/titanic/discussion/28323
- Poncio, F. (2019). Sálvese quien prediga. RPubs by RStudio. https://rstudio-pubs-static.s3.amazonaws.com/555316_3b00cf8efc4c47f4adbc95a4e1f4f1ba.html
- Amat, J. (2018). Machine Learning con R y caret. Cienciadedatos.net. https://www.cienciadedatos.net/documentos/41_machine_learning_con_r_y_caret
- Angkawijaya, A. (2018). Titanic: Machine Learning from Disaster. RStudio pubs. https://rstudio-pubs-static.s3.amazonaws.com/400472_b5699800dc8748608bdef8e555482eaf.html
- Karakasoglou, I. (2020). Exploratory Analysis of the Titanic Dataset. https://jkarakas.github.io/Exploratory-Analysis-of-the-Titanic-Dataset/Titanic_Dataset_Exploratory_Analysis_No_Code.html