Tipología y ciclo de vida. PRACTICA 2: LIMPIEZA Y VALIDACIÓN DE LOS DATOS

Edison Marcelo Muzo Oyana

11 de junio, 2019

Contents

1. Descripción del dataset

```
# Lectura de datos de entrenamiento y prueba.
data_path <- 'input'
train_file <- 'train.csv'
test_file <- 'test.csv'
gender_file <- 'gender_submission.csv'

train_data <- read.csv(paste(data_path, train_file, sep="/"),
                       header = TRUE, stringsAsFactors = FALSE)
test_data <- read.csv(paste(data_path, test_file, sep="/"),
                      header = TRUE, stringsAsFactors = FALSE)
gender_data <- read.csv(paste(data_path, gender_file, sep="/"),
                      header = TRUE, stringsAsFactors = FALSE)
test_data <-merge(x=test_data,y=gender_data,by="PassengerId",all=TRUE)

# Conjunto de datos completo.
full_data  <- bind_rows(train_data, test_data) # bind training & test data
```

El conjunto de datos objeto de análisis se ha obtenido a partir Titanic que contiene datos sobre la supervivencia de pasajeros a bordo del Titanic. Los datos se han dividido en dos grupos:

1. El conjunto de datos de entrenamiento (train.csv). Está constituido por 891 características (columnas) que presentan 12 pasajeros (filas o registros).
2. El conjunto de datos de pruebas (test.csv). Está constituido por 418 características (columnas) que presentan 12 pasajeros (filas o registros).

También se incluye un conjunto de predicciones (gender_submission.csv) que asumen que todos y solo las pasajeras mujeres sobreviven.

Los campos de este conjunto de datos son los siguientes:

Nombre de la Variable	Descripción	Valores
Survived	Survived (1) or died (0)	Survived (1) or died (0)
Pclass	Clase del Pasajero	1 = 1st, 2 = 2nd, 3 = 3rd
Name	Nombre del Pasajero	Caracteres
Sex	Sexo del Pasajero	female or male
Age	Edad del Pasajero	Numérico
SibSp	Número de hermanos / cónyuges a bordo	Numérico
Parch	Número de padres / hijos a bordo	Numérico
Ticket	Número del Ticket	Caracteres
Fare	Tarifa	Caracteres
Cabin	Cabina	Caracteres
Embarked	Puerto de embarque	C = Cherbourg, Q = Queenstown, S = Southampton

Para este trabajo se utilizan los **conjuntos de datos entrenamiento** y **conjunto de datos pruebas** como un solo conjunto de datos. Por tanto, este conjunto de datos contiene 1309 registros y 12 características

Del análisis de los ficheros train.csv y test.csv podemos extraer la siguiente información:

1. Las columnas tienen nombres (nombres de las variables).
2. El separador de columnas es el carácter ***coma*** (,).
3. Las cadenas de caracteres están delimitadas por el carácter ***comilla doble*** (").
4. Algunas cadenas de caracteres tienen espacios en blanco al inicio y/o al final.
5. Los valores decimales tienen el separador decimal ***punto*** (.).
6. El resto de las columnas parecen ser números.

1.2 Importancia y objetivos de los análisis.

A partir de este conjunto de datos se plantea la problemática de determinar qué variables influyeron más sobre la supervivencia de los pasajeros a bordo del Titanic. Además, se podrá proceder a crear modelos de aprendizaje automático que permitan predecir la supervivencia de una persona en función de sus características y contrastes de hipótesis que ayuden a identificar propiedades interesantes en las muestras que puedan ser inferidas con respecto a la población.

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

En primer lugar, inspeccionamos el conjunto de datos sin ningún tipo de pre-procesamiento, para ello se utiliza la función **sr()**.

```
# Visualizamos los datos cargados
str(full_data)
```

```
## 'data.frame':    1309 obs. of  12 variables:
```

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

De este conjunto de datos extraemos las siguientes conclusiones:

1. La característica `PassengerId` se puede eliminar del conjunto de datos ya que no contribuye a la supervivencia.
2. La característica `Ticket` también se puede eliminar del conjunto de datos ya que no parece contribuir a la supervivencia.
3. De la característica `Name` se puede extraer el título (por ejemplo, 'Miss', 'Mrs', etc) y el apellido de la familia y pueden aportar información adicional para determinar la supervivencia.
4. De la característica `Cabina` se pueden crear grupos según la letra inicial de la cabina y pueden aportar información adicional para determinar la supervivencia. En los casos que un valor tenga múltiples cabinas a proiri parecen compartir la misma letra y solo cambia el número de cabina, así que también nos quedamos con la primera letra.
5. De las características `SibSp` y `Parch` se puede combinar para obtener el tamaño de la familia y puede aportar información adicional para determinar la supervivencia.

El resto las características (`Pclass`, `Sex`, `Age`, `SibSp`, `Parch`, `Fare` y `Embarked`) del conjunto de datos serán considerados durante la realización de los análisis.

3. Limpieza de los datos

El conjunto de datos (train + test) contiene 1309 registros y 12 variables. Los nombres de las características son: PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked. Antes de comenzar con la tarea de la limpieza de los datos vamos a identificar los **tipos de datos de variables**, para ello se puede usar las funciones `str()` o `glimpse()`. Para mostrar esta información en forma de tabla que facilita el análisis, se utiliza la función `sapply(dataset, class)`.

```
# Inspeccionamos la estructura del conjunto de datos
str(full_data)
```

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

```r
# Inspeccionamos el conjunto de datos
glimpse(full_data)
```

```
## Observations: 1,309
## Variables: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,...
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3,...
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bra...
## $ Sex         <chr> "male", "female", "female", "female", "male", "mal...
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, ...
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4,...
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1,...
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "1138...
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, ...
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", ...
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", ...
```

```r
# Mostramos en forma de tabla
column_classes <- sapply(full_data, class)
data <- data.frame(Variables = names(column_classes), Clases=unname(column_classes))
kable(data) %>%
    kable_styling(bootstrap_options = "striped", full_width = F)
```

Variables	Clases
PassengerId	integer
Survived	integer
Pclass	integer
Name	character
Sex	character
Age	numeric
SibSp	integer
Parch	integer
Ticket	character
Fare	numeric
Cabin	character
Embarked	character

A la vista de los resultados anteriores se identifican las siguientes conversiones:

- La característica `Survived` debería ser un factor debido a que es cualitativa con dos valores: `1` y `0`.
- La característica `PClass` debería ser un factor debido a que es cualitativa con tres valores: `1`, `2` y `3`.
- La característica `Sex` debería ser un factor debido a que es cualitativa con dos valores: `male` y `female`.
- La característica `Embarked` debería ser un factor debido a que es cualitativa con tres valores: `C`, `Q`, y `S`. Además, hay que cambiar los valores vacíos a `NA`.
- En la característica `Cabin` hay que cambiar los valores vacíos a `NA`.

Además, se requiere extraer información de las siguientes características:

- De la característica `Name` se extraer el título y el apellido de la familia.
- De la característica `Cabina` se extrae el grupo de la cabina.
- De las características `SibSp` y `Parch` se combinan para obtener el tamaño de la familia.

Conversiones

En primer lugar, convertimos a factores las características `Survived`, `PClass`, `Sex` y `Embarked`. Convertimos los valores vacíos a `NA` en las características `Embarked` y `Cabin`. Finalmente, visualizamos los tipos de las características para comprobar las conversiones.

```
# Conversion a Factores
full_data$Survived <- as.factor(full_data$Survived)
full_data$Pclass <- as.factor(full_data$Pclass)
full_data$Sex <- as.factor(str_to_upper(str_trim(full_data$Sex)))
levels(full_data$Sex)
```

```
## [1] "FEMALE" "MALE"
```

```
levels(full_data$Sex) <- c("F", "M")
full_data$Embarked <- factor(full_data$Embarked, exclude = '')


# Conversion de vacios a NA.
full_data$Cabin <- str_trim(full_data$Cabin)
full_data$Cabin[full_data$Cabin == ''] <- NA
full_data$Ticket <- str_trim(full_data$Ticket)
full_data$Ticket[full_data$Ticket == ''] <- NA


# Mostratamos el resultado de las conversiones:
str(full_data)
```

```
## 'data.frame':    1309 obs. of  12 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 "F","M": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  NA "C85" NA "C123" ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

```
# Visualizamos la tabla
column_classes <- sapply(full_data, class)
data <- data.frame(Variables = names(column_classes), Clases=unname(column_classes))
kable(data) %>%
    kable_styling(bootstrap_options = "striped", full_width = F)
```

Variables	Clases
PassengerId	integer
Survived	factor
Pclass	factor
Name	character
Sex	factor
Age	numeric
SibSp	integer
Parch	integer
Ticket	character
Fare	numeric
Cabin	character
Embarked	factor

Después de estas transformaciones tenemos la siguiente distribución de variables:

- Variables categóricas: `Survived`, `Sex`, `Embarked`, y `Pclass`.
- Variables numéricas continuas: `Age`, `Fare.`
- Variables numéricas discretas: `SibSp`, `Parch.`
- Variables con caracteres: `Name`, `Ticket` y `Cabin`.
  - `Name`: Caracteres alfanuméricos.
  - `Ticket`: Mezcla de caracteres especiales y alfanuméricos.
  - `Cabin`: Caracteres alfanuméricos.

Característica Nombre (Name)

La variable nombre del pasajero podemos dividirla en variables significativas adicionales que pueden alimentar predicciones o ser usadas en la creación de nuevas variables adicionales. Por ejemplo, el título del pasajero está contenido dentro de la variable de nombre del pasajero (Por ejemplo, 'Mr', 'Miss') y podemos usar el apellido para representar a las familias.

```
# Grab title from passenger names
full_data$Title <- gsub('(.*, )|(\\..*)', '', full_data$Name)

# Show title counts by sex
table(full_data$Sex, full_data$Title)
```

```
##
##     Capt Col Don Dona  Dr Jonkheer Lady Major Master Miss Mlle Mme  Mr Mrs
##   F    0   0   0    1   1        0    1     0      0  260    2   1   0 197
##   M    1   4   1    0   7        1    0     2     61    0    0   0 757   0
##
##      Ms Rev Sir the Countess
##   F   2   0   0            1
##   M   0   8   1            0
```

```
# Titles with very low cell counts to be combined to "rare" level
rare_title <- c('Dona', 'Lady', 'the Countess','Capt', 'Col', 'Don',
                'Dr', 'Major', 'Rev', 'Sir', 'Jonkheer')

# Also reassign mlle, ms, and mme accordingly
full_data$Title[full_data$Title == 'Mlle']        <- 'Miss'
full_data$Title[full_data$Title == 'Ms']          <- 'Miss'
full_data$Title[full_data$Title == 'Mme']         <- 'Mrs'
full_data$Title[full_data$Title %in% rare_title]  <- 'Rare Title'
```

```
# Conversion a factor
full_data$Title <- as.factor(full_data$Title)

# Show title counts by sex again
table(full_data$Sex, full_data$Title)
```

```
##
##     Master Miss  Mr Mrs Rare Title
##   F      0  264   0 198          4
##   M     61    0 757   0         25
```

```
# Finally, grab surname from passenger name
full_data$Surname <- sapply(full_data$Name,
                      function(x) strsplit(x, split = '[,.]')[[1]][1])

# Conversion a factor
full_data$Surname <- as.factor(full_data$Surname)
```

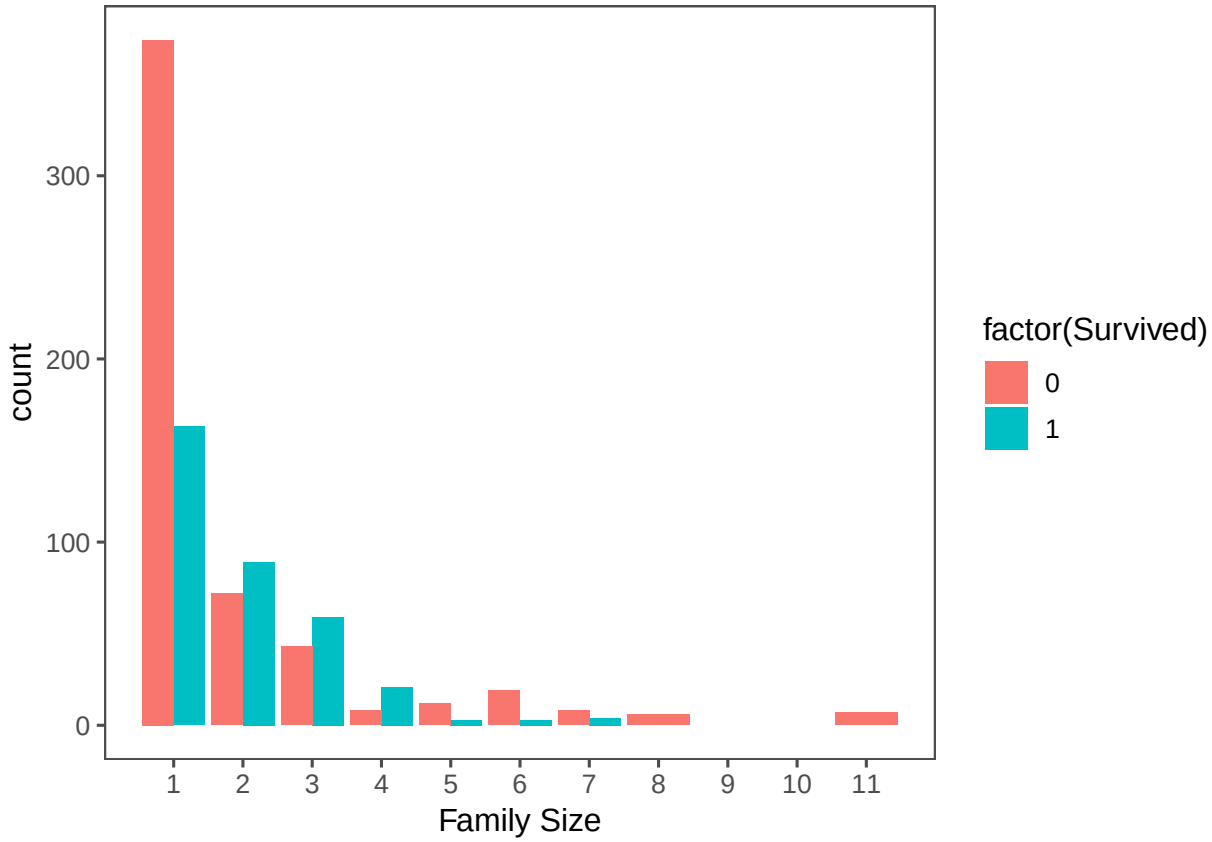
Característica Tamaño de la familia.

Podemos combinar los valores de las características `SibSp` y `Parch` para crear una característica discreta con el tamaño de la variable `FsizeD`.

```
# Create a family size variable including the passenger themselves
full_data$Fsize <- full_data$SibSp + full_data$Parch + 1
```

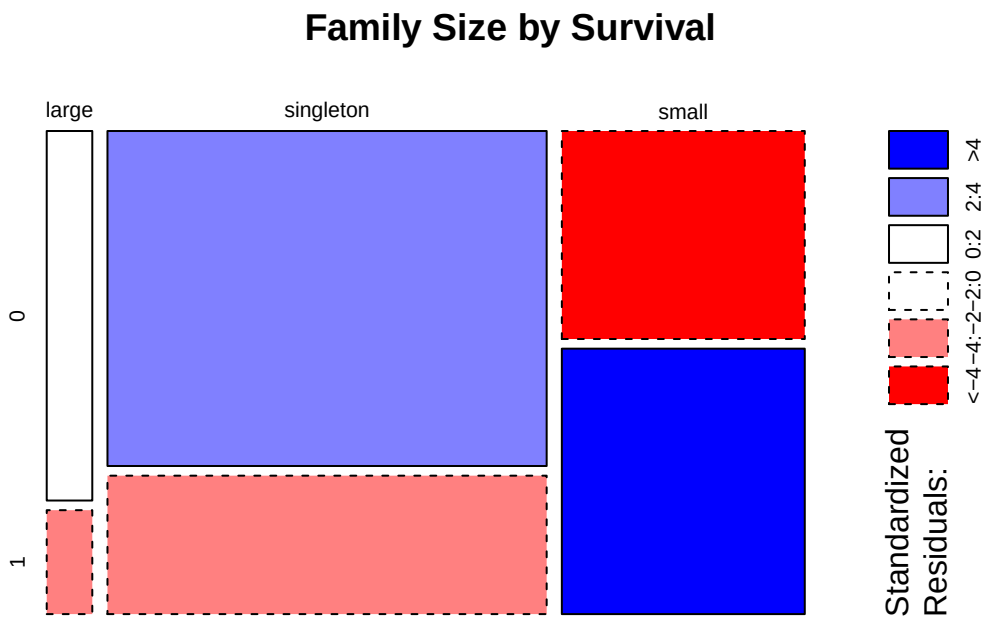
Visualizamos la posible relación entre el tamaño de la familia y la supervivencia.

```
# Use ggplot2 to visualize the relationship between family size & survival
ggplot(full_data[1:891,], aes(x = Fsize, fill = factor(Survived))) +
  geom_bar(stat='count', position='dodge') +
  scale_x_continuous(breaks=c(1:11)) +
  labs(x = 'Family Size') +
  theme_few()
```



Dado los resultados anteriores, podemos observar que hay una penalización de supervivencia para los solteros y aquellos con un tamaño de familia superior a 4. Se puede discretizar esta variable en tres niveles, lo que será útil ya que hay comparativamente menos familias grandes.

```
# Discretize family size
full_data$FsizeD[full_data$Fsize == 1] <- 'singleton'
full_data$FsizeD[full_data$Fsize < 5 & full_data$Fsize > 1] <- 'small'
full_data$FsizeD[full_data$Fsize > 4] <- 'large'

full_data$FsizeD <- as.factor(full_data$FsizeD)

# Show family size by survival using a mosaic plot
mosaicplot(table(full_data$FsizeD, full_data$Survived), main='Family Size by Survival', shade=TRUE)
```



Característica Cabina (Cabin)

De la variable cabina (**Cabine**) podemos extraer alguna información potencialmente útil. Para ello se va a discretizar esta variable según la primera letra de la cabina. Existen registros donde la cabina tiene múltiples valores, pero a priori en estos casos la letra inicial de la cabina es la misma variando el número.

```
# This variable appears to have a lot of missing values
head(full_data)$Cabin
```

```
## [1] NA     "C85"  NA     "C123" NA     NA
```

```
# The first character is the deck. For example:
strsplit(full_data$Cabin[2], NULL)[[1]]
```

```
## [1] "C" "8" "5"
```

```
# Create a Deck variable. Get passenger deck A - F:
full_data$Deck<- sapply(full_data$Cabin, function(x) strsplit(x, NULL)[[1]][1])
```

Característica Ticket

De la variable ticket (**Ticket**) podemos extraer alguna información potencialmente útil. Varios pasajeros están asociados a un ticket. Para ello se va a eliminar caracteres no alfanuméricos y se transformarán en factores sus valores.

```
# Eliminamos el punto y la barra inclinada
full_data$Ticket <- gsub('\\.|/|\\s', "", full_data$Ticket)
```

```r
# Convertimos en facto
full_data$Ticket <- as.factor(full_data$Ticket)
```

3.1 Ceros o elementos vacíos

Para analizar las características con valores nulos e incompletos visualizamos un resumen de los variables con la función `summary()`:

```r
summary(full_data)
```

```
##   PassengerId   Survived Pclass      Name              Sex
##  Min.   :   1   0:815    1:323   Length:1309        F:466
##  1st Qu.: 328   1:494    2:277   Class :character   M:843
##  Median : 655            3:709   Mode  :character
##  Mean   : 655
##  3rd Qu.: 982
##  Max.   :1309
##
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.000   CA2343 :  11
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000   1601   :   8
##  Median :28.00   Median :0.0000   Median :0.000   CA2144 :   8
##  Mean   :29.88   Mean   :0.4989   Mean   :0.385   3101295:   7
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000   347077 :   7
##  Max.   :80.00   Max.   :8.0000   Max.   :9.000   347082 :   7
##  NA's   :263                                      (Other):1261
##       Fare            Cabin           Embarked          Title
##  Min.   :  0.000   Length:1309        C   :270   Master    : 61
##  1st Qu.:  7.896   Class :character   Q   :123   Miss      :264
##  Median : 14.454   Mode  :character   S   :914   Mr        :757
##  Mean   : 33.295                      NA's:  2   Mrs       :198
##  3rd Qu.: 31.275                                 Rare Title: 29
##  Max.   :512.329
##  NA's   :1
##       Surname         Fsize              FsizeD        Deck
##  Andersson:  11   Min.   : 1.000   large    : 82   Length:1309
##  Sage     :  11   1st Qu.: 1.000   singleton:790   Class :character
##  Asplund  :   8   Median : 1.000   small    :437   Mode  :character
##  Goodwin  :   8   Mean   : 1.884
##  Davies   :   7   3rd Qu.: 2.000
##  Brown    :   6   Max.   :11.000
##  (Other)  :1258
```

```r
# Visualizar numero de nulos en las variables.
mv_colnames <- colSums(is.na(full_data))
mv_colnames <- mv_colnames[mv_colnames > 0]
data <- data.frame(Variables = names(mv_colnames), Missing=unname(mv_colnames))
kable(data) %>%
    kable_styling(bootstrap_options = "striped", full_width = F)
```

Variables	Missing
Age	263
Fare	1
Cabin	1014
Embarked	2
Deck	1014

Dado los resultado anteriores la características

- Los valores **NA** de la característica **Survived** de deben a que es característica no esta en el conjunto de datos de prueba.
- La característica **Deck** se ha extraído a pator de los valores de la característica **Cabin**.

Por tanto, las variables de interés que tienen valores perdidos ordenadas de mayor a menor son: `Deck` > `Age` > `Embarked` > `Fare`.

Característica Embarque

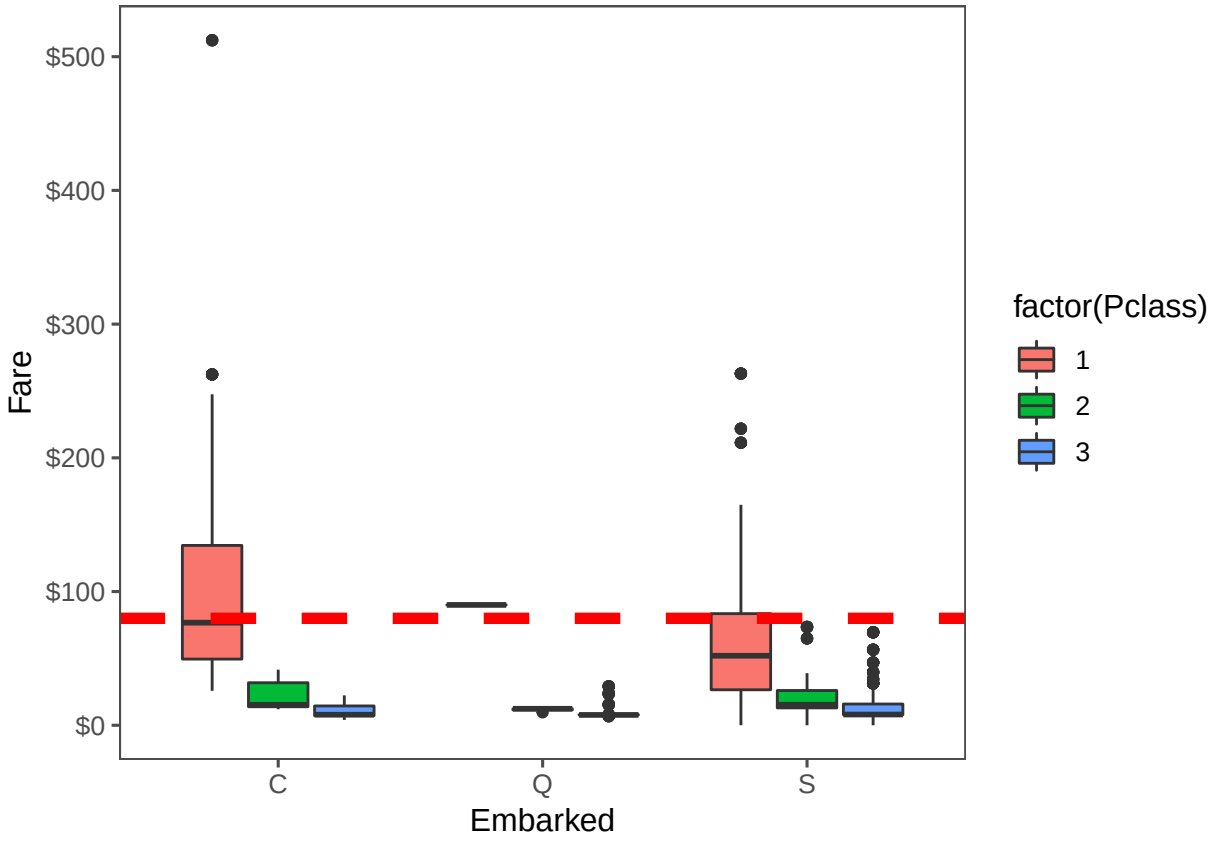
Antes de imputar los valores perdidos de la característica Embarque (**Embarked**) visualizamos los datos que tienen valores perdidos para esta característica ya que son pocos.

```
# Passengers 62 and 830 are missing Embarkment

miss_embark_index <- which(is.na(full_data$Embarked))
miss_embark <- full_data[miss_embark_index,]
miss_embark
```

```
##     PassengerId Survived Pclass                                         Name
## 62           62        1      1                        Icard, Miss. Amelie
## 830         830        1      1 Stone, Mrs. George Nelson (Martha Evelyn)
##     Sex Age SibSp Parch Ticket Fare Cabin Embarked Title Surname Fsize
## 62    F  38     0     0 113572   80   B28     <NA>  Miss   Icard     1
## 830   F  62     0     0 113572   80   B28     <NA>   Mrs   Stone     1
##        FsizeD Deck
## 62  singleton    B
## 830 singleton    B
```

Podemos inferir sus valores de embarque en función de los datos actuales que podamos imaginar que pueden ser relevantes: clase de pasajero y tarifa. Se observa que ambos pagaron $ 80 y estaban en la clase 1.

```
# Get rid of our missing passenger IDs
embark_fare <- full_data %>%
  filter(!is.na(Embarked))

# Use ggplot2 to visualize embarkment, passenger class, & median fare
ggplot(embark_fare, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +
  geom_boxplot() +
  geom_hline(aes(yintercept=80),
    colour='red', linetype='dashed', lwd=2) +
  scale_y_continuous(labels=dollar_format()) +
  theme_few()
```



Dado los resultados anteriores, se observa que la tarifa mediana para un pasajero de 1ra clase que sale de Charbourg ('C') coincide muy bien con los $ 80 pagados por los pasajeros con valores perdidos en el embarque. Por tanto, podemos asignarles el valor 'C'.

```
# Since their fare was $80 for 1st class, they most likely embarked from 'C'
full_data$Embarked[miss_embark_index] <- 'C'

# Comprobamos el resultado
sum(is.na(full_data$Embarked))
```

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

Característica Tarifa

Antes de imputar los valores perdidos de la característica Embarque (**Fare**) visualizamos los datos que tienen valores perdidos para esta característica ya que son pocos.

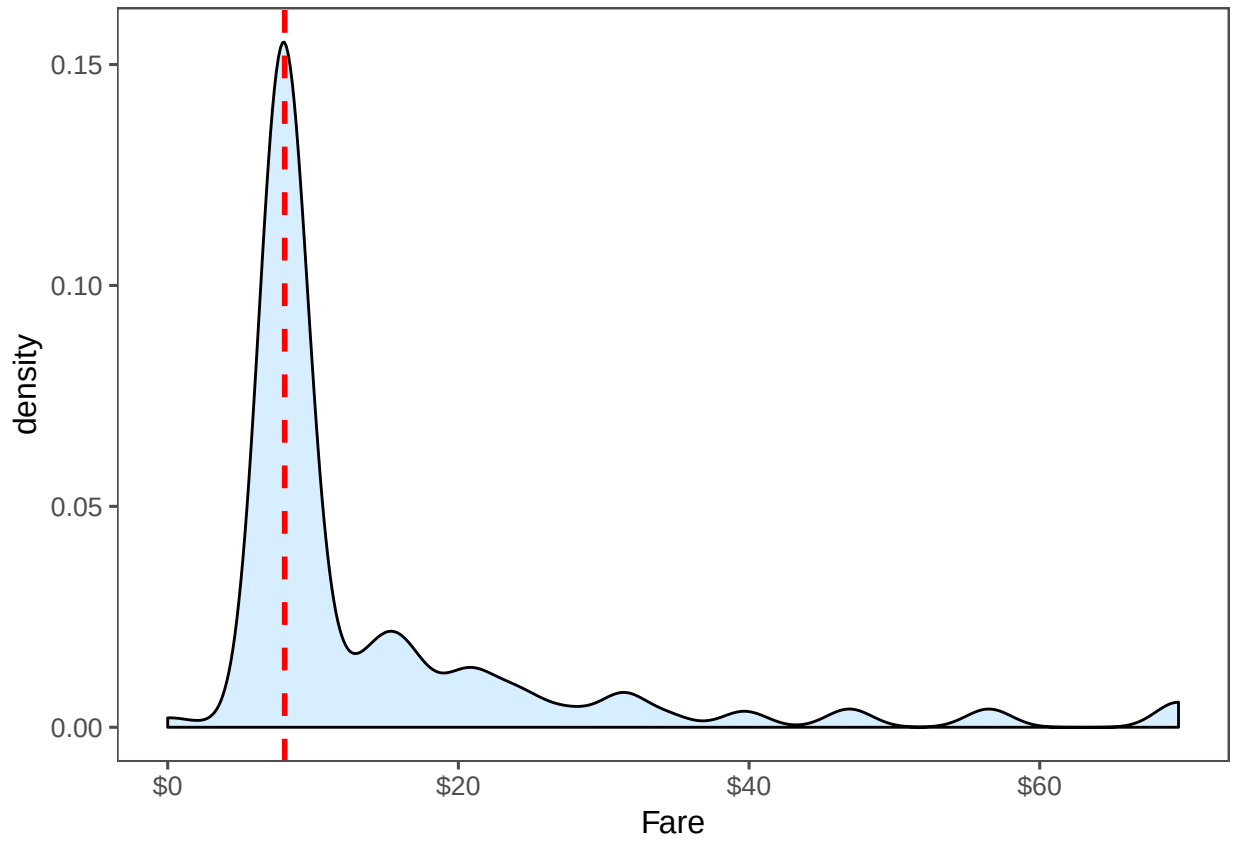
```
# Show row 1044
miss_fare_index <- which(is.na(full_data$Fare))
miss_fare <- full_data[miss_fare_index,]
miss_fare
```

```
##      PassengerId Survived Pclass               Name Sex  Age SibSp Parch
## 1044        1044        0      3 Storey, Mr. Thomas   M 60.5     0     0
##      Ticket Fare Cabin Embarked Title Surname Fsize    FsizeD Deck
## 1044   3701   NA  <NA>        S    Mr  Storey     1 singleton <NA>
```

Dado los resultados anteriores, se observa que el pasajero estaba asignado a la tercera clase y que partió de Southampton ("S"). Ahora, visualizamos las tarifas entre todos los demás que comparten su clase y su

embarque (n = 495).

```
# Get rid of our missing passenger IDs
pclass_embark <- full_data %>%
  filter(Pclass == '3' & Embarked == 'S')

ggplot(pclass_embark,
  aes(x = Fare)) +
  geom_density(fill = '#99d6ff', alpha=0.4) +
  geom_vline(aes(xintercept=median(Fare, na.rm=T)),
    colour='red', linetype='dashed', lwd=1) +
  scale_x_continuous(labels=dollar_format()) +
  theme_few()
```



Dado los resultados obtenidos, parece bastante razonable reemplazar el valor perdido de la tarifa por la mediana de su clase y embarque, que es de $ 8.05.

```
# Replace missing fare value with median fare for class/embarkment
full_data$Fare[miss_fare_index] <- median(full_data[full_data$Pclass == '3' & full_data$Embarked == 'S'

# Comprobamos el resultado
sum(is.na(full_data$Fare))
```

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

Característica Deck

Para la característica **Deck** se observa que existen muchos valores perdidos. No parece un buen método usar la media o la mediana para inferir los valores más probable. Sin embargo, sabemos que el valor de la característica **Deck** esta relacionada con la clase del pasajero. Por tanto, se asignará un valor que representen esta falta de información en función de clase del pasajero. Esta asignación será de la siguiente forma:

- *U1*. Para pasajeros de primera clase.
- *U2*. Para pasajeros de segunda clase.
- *U3*. Para pasajeros de tercera clase.

```
# Remplazamos los valores perdidos de
miss_deck_index <- which(full_data$Pclass == '1' & is.na(full_data$Deck))
full_data$Deck[miss_deck_index] <- 'U1'
miss_deck_index <- which(full_data$Pclass == '2' & is.na(full_data$Deck))
full_data$Deck[miss_deck_index] <- 'U2'
miss_deck_index <- which(full_data$Pclass == '3' & is.na(full_data$Deck))
full_data$Deck[miss_deck_index] <- 'U3'

full_data$Deck <- factor(full_data$Deck)

# Comprobamos el resultado
sum(is.na(full_data$Deck))
```

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

Característica Edad

Finalmente, la variable Edad (**Age**) tiene bastantes valores perdidos. Para calcular los valores perdidos se utiliza un modelo de predicción más sofisticado basado en otras variables, llamado mice (Multivariate Imputation by Chained Equations).

Este método se basa en la **Especificación Totalmente Condicional**, donde cada variable incompleta se imputa por un modelo separado. El algoritmo MICE puede imputar mezclas de datos categóricos ordenados, continuos, binarios, desordenados. Además, MICE puede imputar datos continuos de dos niveles y mantener la coherencia entre las imputaciones mediante la imputación pasiva.

```
# Make variables factors into factors

# Set a random seed
set.seed(129)

# Perform mice imputation, excluding certain less-than-useful variables:
mice_mod <- mice(full_data[, !names(full_data) %in%
  c('PassengerId','Name','Ticket','Cabin','Family','Surname','Survived')],
  method='rf')
```

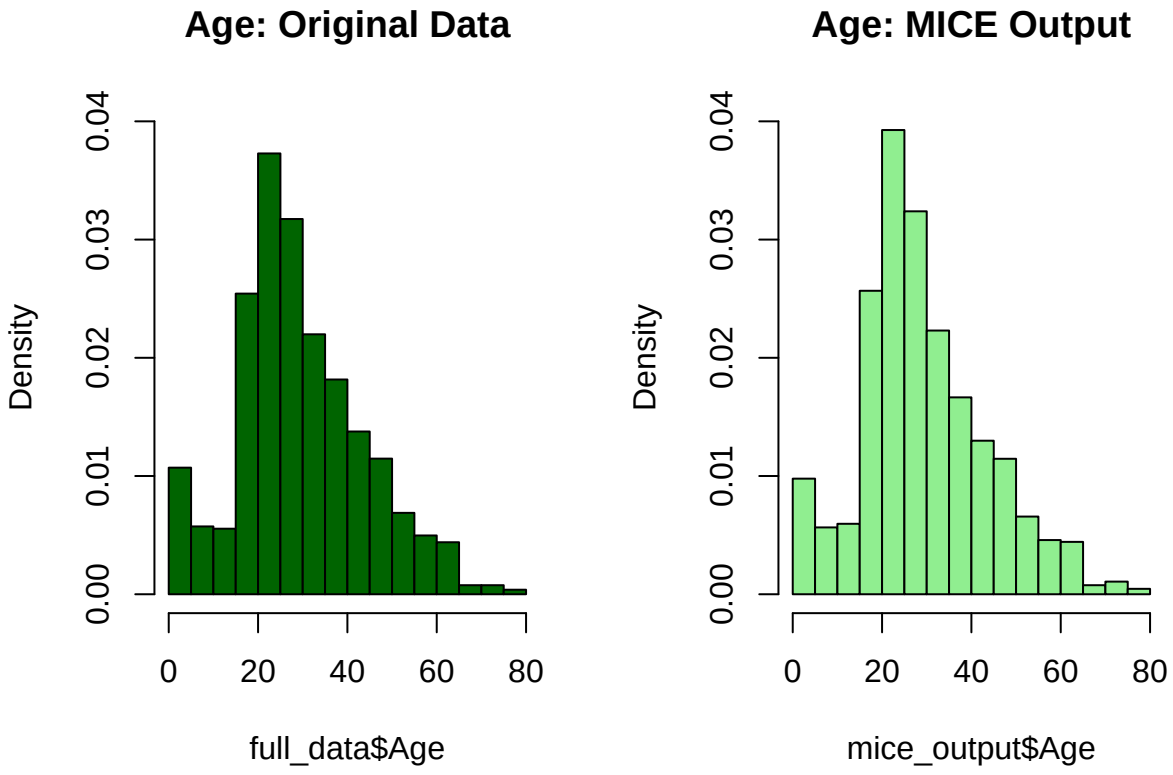
```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
```

```
##   2   4  Age
##   2   5  Age
##   3   1  Age
##   3   2  Age
##   3   3  Age
##   3   4  Age
##   3   5  Age
##   4   1  Age
##   4   2  Age
##   4   3  Age
##   4   4  Age
##   4   5  Age
##   5   1  Age
##   5   2  Age
##   5   3  Age
##   5   4  Age
##   5   5  Age
```

```r
# Save the complete output
mice_output <- complete(mice_mod)
```

Comparamos los resultados de la distribución original de la edad con los del modelo.

```r
# Plot age distributions
par(mfrow=c(1,2))
hist(full_data$Age, freq=F, main='Age: Original Data',
  col='darkgreen', ylim=c(0,0.04))
hist(mice_output$Age, freq=F, main='Age: MICE Output',
  col='lightgreen', ylim=c(0,0.04))
```



Dado los resultados anteriores, se observa una leve mejora en la distribución. Por tanto, se remplaza los datos originales de la edad con los obtenidos con el modelo `mice`.

```
# Replace Age variable from the mice model.
full_data$Age <- mice_output$Age

# Show new number of missing Age values
sum(is.na(full_data$Age))
```

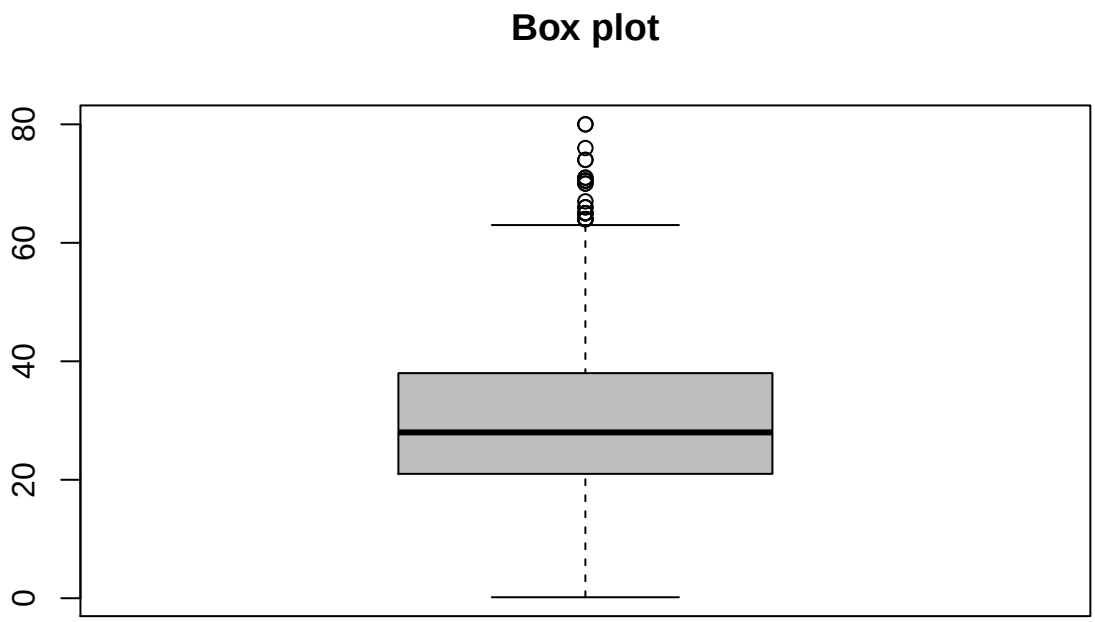
```
## [1] 0
```

3.2. Identificación y tratamiento de valores extremos

Los valores extremos o **outliers** son aquellos que parecen no ser congruentes sin los comparamos con el resto de los datos. Para identificarlos se representará un diagrama de caja por cada variable y ver qué valores distan mucho del rango intercuartílico (la caja), para ello se utilizará la función `boxplots.stats()`.

Así, se mostrarán sólo los valores atípicos para variables cuantitativas: `Age`, `Fare`, `SibSp`, `Parch`, y `Fsize`.

```
# Visualizamos boxplot
boxplot(full_data$Age, main="Box plot", col="gray")
```

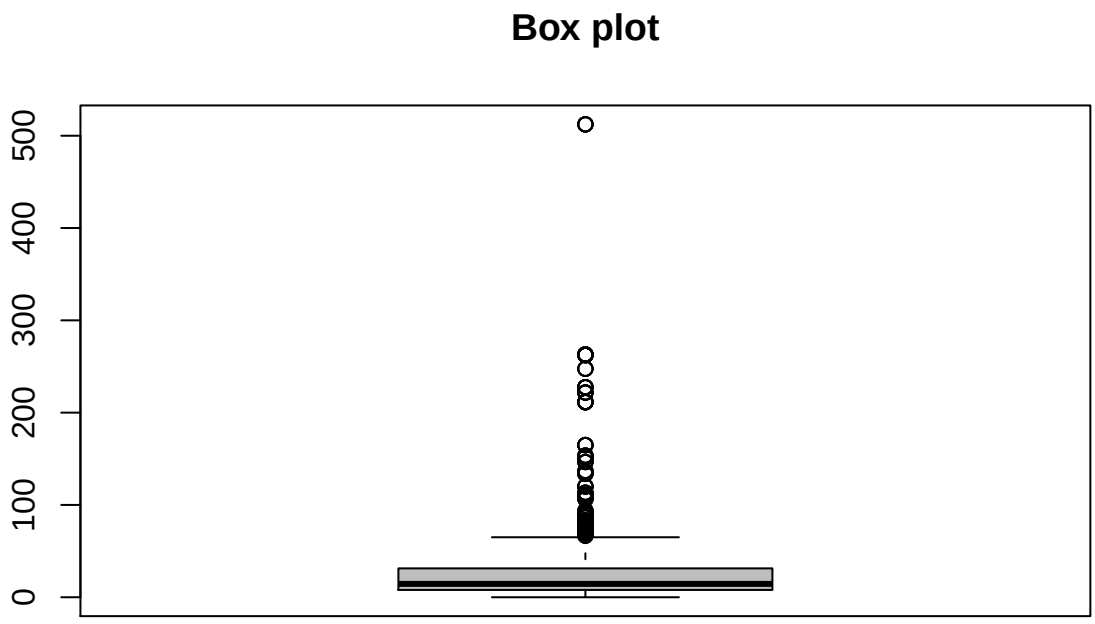



```
boxplot.stats(full_data$Age)$out
```

```
##  [1] 70.5 66.0 65.0 71.0 70.5 66.0 80.0 65.0 65.0 64.0 65.0 70.5 71.0 64.0
## [15] 80.0 70.0 70.0 74.0 74.0 67.0 76.0 64.0 64.0 64.0
```

Para los resultados de la característica **Edad**, si revisamos de forma aleatoria los datos de los pasajeros se comprueba que los valores extremos están un rango normal. Por ejemplo, ninguno es menor que cero o mayor que 100. Un pasajero con 100 años viajando es poco usual. Por tanto, son valores que perfectamente pueden darse.

```
# Visualizamos boxplot
boxplot(full_data$Fare, main="Box plot", col="gray")
```

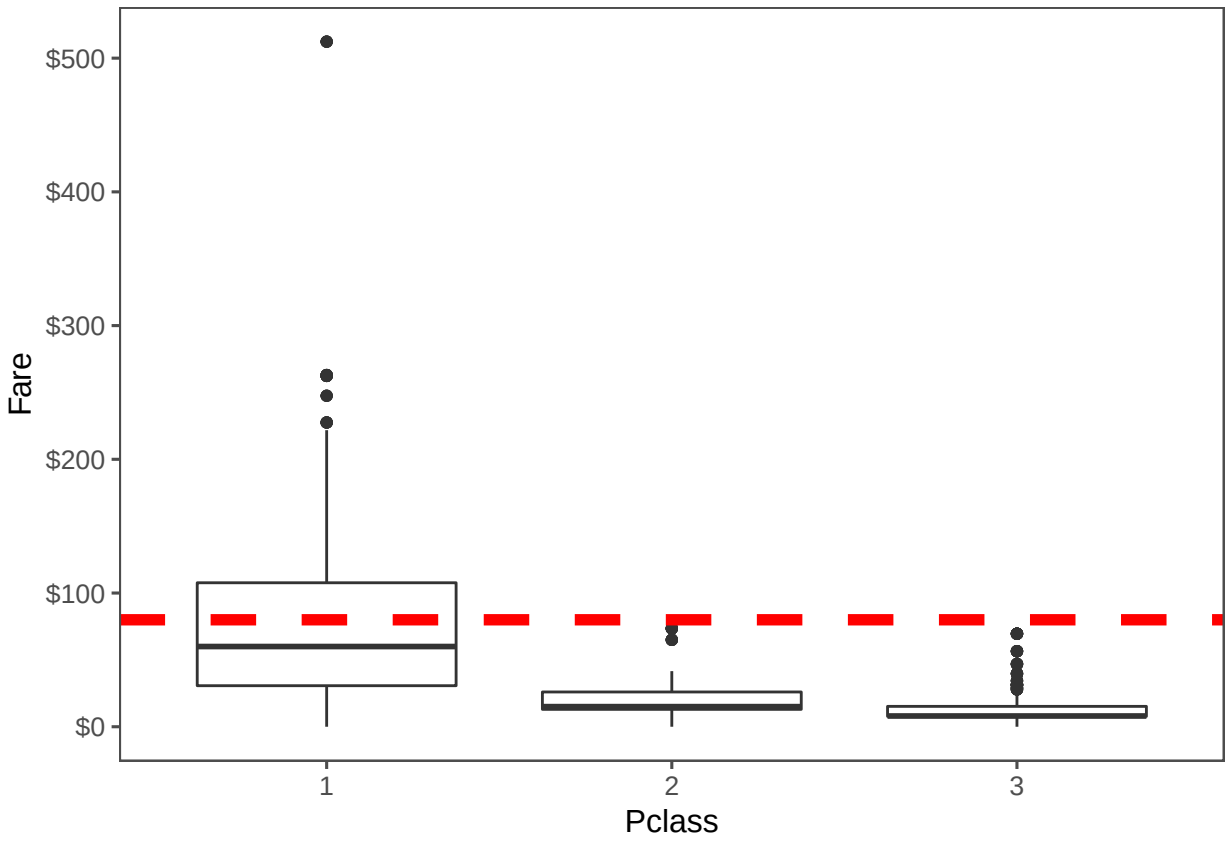



```
boxplot.stats(full_data$Fare)$out
```

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

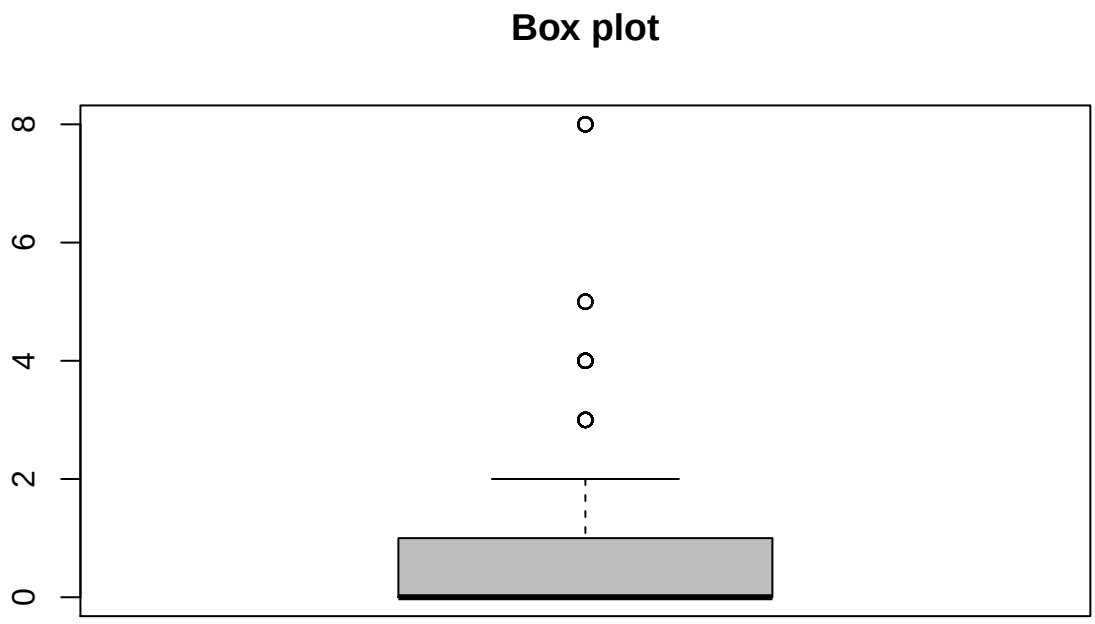
```
## [169] 211.5000  90.0000 108.9000
```

```
# Use ggplot2 to visualize Pclass, passenger class, & median fare
ggplot(full_data, aes(x = Pclass, y = Fare)) +
  geom_boxplot() +
  geom_hline(aes(yintercept=80),
    colour='red', linetype='dashed', lwd=2) +
  scale_y_continuous(labels=dollar_format()) +
  theme_few()
```



Para los resultados de la característica **Tarifa**, si revisamos de forma aleatoria los datos de los pasajeros se comprueba que los valores extremos están asociados a un mismo ticket en una clase de pasajero especifica. Mientras mejor es la clase y mayor es el número de pasajeros, más alta es la tarifa. Por tanto, son valores que perfectamente pueden darse.

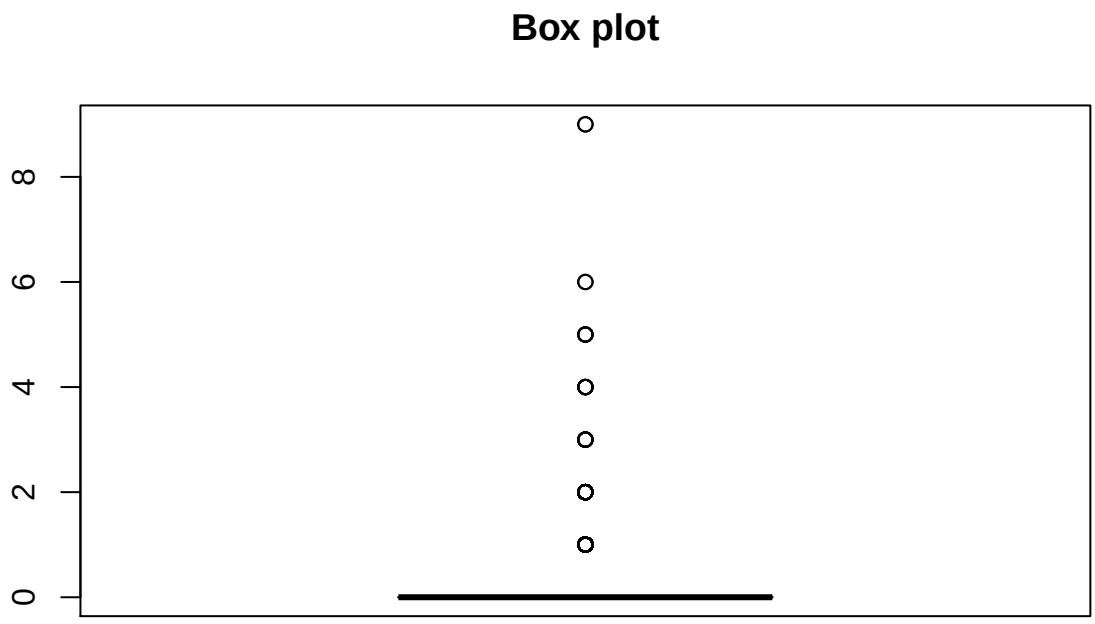
```
boxplot(full_data$SibSp, main="Box plot", col="gray")
```




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

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

```
boxplot(full_data$Parch, main="Box plot", col="gray")
```




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

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

Para los resultados de las características **Número de hermanos / cónyuges a bordo (SibSp)** y **Número de padres / hijos a bordo(Parch)**; si revisamos de forma aleatoria los datos de los pasajeros se comprueba que los valores extremos están un rango normal. Por ejemplo, ninguno es menor que cero o mayor que 15. Una familia con más de 20 individuos viajando junto es poco habitual. Por tanto, son valores que perfectamente pueden darse.

3.3. Exportación de los datos preprocesados

Volvemos a revisar las características un vez más con la función `summary()`.

```
summary(full_data)
```

```
##   PassengerId   Survived Pclass      Name              Sex
##  Min.   :   1   0:815    1:323   Length:1309        F:466
##  1st Qu.: 328   1:494    2:277   Class :character   M:843
##  Median : 655            3:709   Mode  :character
```

```
##  Mean   : 655
##  3rd Qu.: 982
##  Max.   :1309
##
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.000   CA2343 :  11
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000   1601   :   8
##  Median :28.00   Median :0.0000   Median :0.000   CA2144 :   8
##  Mean   :29.73   Mean   :0.4989   Mean   :0.385   3101295:   7
##  3rd Qu.:38.00   3rd Qu.:1.0000   3rd Qu.:0.000   347077 :   7
##  Max.   :80.00   Max.   :8.0000   Max.   :9.000   347082 :   7
##                                                   (Other):1261
##       Fare            Cabin           Embarked        Title
##  Min.   :  0.000   Length:1309        C:272    Master    : 61
##  1st Qu.:  7.896   Class :character   Q:123    Miss      :264
##  Median : 14.454   Mode  :character   S:914    Mr        :757
##  Mean   : 33.276                               Mrs       :198
##  3rd Qu.: 31.275                               Rare Title: 29
##  Max.   :512.329
##
##       Surname         Fsize              FsizeD         Deck
##  Andersson:  11   Min.   : 1.000   large    : 82   U3     :693
##  Sage     :  11   1st Qu.: 1.000   singleton:790   U2     :254
##  Asplund  :   8   Median : 1.000   small    :437   C      : 94
##  Goodwin  :   8   Mean   : 1.884                   U1     : 67
##  Davies   :   7   3rd Qu.: 2.000                   B      : 65
##  Brown    :   6   Max.   :11.000                   D      : 46
##  (Other)  :1258                                    (Other): 90
```

De la información anterior se concluye:

- La variable `PassengerId` se puede eliminarse del conjunto de datos ya que no contribuye a la supervivencia.
- La variable `Name` se puede eliminar debido a que se ha extraído su información en las caracteristcias `Title` y `Surname`.
- La variable `Cabin` se puede eliminar debido a que se ha extraído su información en la `Deck`.
- La variable `Fsize` se puede eliminar debido a que uso como una combinación `SibSp` y `Parch`.

Por tanto, se seleccionan las siguientes caracteristicas: `Age`, `Sex`, `SibSp`, `Parch`, `Pclass`, `Fare`, `Ticket`, `Title`, `Surname`, `Deck`, y `FSizeD`.

```
# Seleccion de caracteristicas de interes
#cleaning_full_data <- select(full_data, -PassengerId, -Name, -Cabin, -Fsize)
cleaning_full_data <- select(full_data, -Name, -Cabin, -Fsize)

# Visualizamos los datos limpios:
summary(cleaning_full_data)
```

```
##   PassengerId   Survived Pclass  Sex          Age            SibSp
##  Min.   :   1   0:815    1:323   F:466   Min.   : 0.17   Min.   :0.0000
##  1st Qu.: 328   1:494    2:277   M:843   1st Qu.:21.00   1st Qu.:0.0000
##  Median : 655            3:709           Median :28.00   Median :0.0000
##  Mean   : 655                            Mean   :29.73   Mean   :0.4989
##  3rd Qu.: 982                            3rd Qu.:38.00   3rd Qu.:1.0000
##  Max.   :1309                            Max.   :80.00   Max.   :8.0000
##
```

```
##      Parch           Ticket          Fare         Embarked
##  Min.   :0.000   CA2343 :  11   Min.   :  0.000   C:272
##  1st Qu.:0.000   1601   :   8   1st Qu.:  7.896   Q:123
##  Median :0.000   CA2144 :   8   Median : 14.454   S:914
##  Mean   :0.385   3101295:   7   Mean   : 33.276
##  3rd Qu.:0.000   347077 :   7   3rd Qu.: 31.275
##  Max.   :9.000   347082 :   7   Max.   :512.329
##                  (Other):1261
##         Title          Surname           FsizeD         Deck
##  Master    : 61   Andersson:  11   large    : 82   U3     :693
##  Miss      :264   Sage     :  11   singleton:790   U2     :254
##  Mr        :757   Asplund  :   8   small    :437   C      : 94
##  Mrs       :198   Goodwin  :   8                   U1     : 67
##  Rare Title: 29   Davies   :   7                   B      : 65
##                   Brown    :   6                   D      : 46
##                   (Other)  :1258                   (Other): 90
```

```
# Dividimos el conjunto de datos en datos de entrenamiento y datos prueba.
cleaning_train_data <- cleaning_full_data[1:nrow(train_data),]
cleaning_test_data <- cleaning_full_data[(nrow(train_data) + 1):nrow(full_data),]
#cleaning_test_data <- select(cleaning_test_data, -Survived)

# Exportación de los datos limpios en .csv
output_path <- 'output'
cleaning_train_file <- 'cleaning_train.csv'
cleaning_test_file <- 'cleaning_test.csv'
cleaning_full_file <- 'cleaning_full.csv'

write.csv(cleaning_train_data,
          paste(output_path, cleaning_train_file, sep = '/'),
          quote = FALSE, row.names=F)
write.csv(cleaning_test_data,
          paste(output_path, cleaning_test_file, sep = '/'),
          quote = FALSE, row.names=F)
write.csv(cleaning_full_data,
          paste(output_path, cleaning_full_file, sep = '/'),
          quote = FALSE, row.names=F)
```

Dividimos el conjunto de datos limpios en dos conjuntos:

- El conjunto de **datos de entrenamiento limpios** se almacena en el fichero cleaning_train.csv y está constituido por 891 características y 12 pasajeros.
- El conjunto de **datos de pruebas limpios** se almacena en el fichero cleaning_test.csv y está constituido por 418 características y 12 pasajeros.

4. Análisis de los datos.

```
# Lectura de datos de entrenamiento y prueba.
output_path <- 'output'
data_path <- 'input'
cleaning_train_file <- 'cleaning_train.csv'
cleaning_test_file <- 'cleaning_test.csv'
cleaning_full_file <- 'cleaning_full.csv'
gender_file <- 'gender_submission.csv'
```

```
clean_train <- read.csv(paste(output_path, cleaning_train_file, sep="/"),
                        header = TRUE, stringsAsFactors = FALSE)
clean_test <- read.csv(paste(output_path, cleaning_test_file, sep="/"),
                       header = TRUE, stringsAsFactors = FALSE)


# Conjunto de datos completo.
cleaning_full_data  <- bind_rows(clean_train, clean_test) # bind training & test data

clean_train$Survived <- as.factor(clean_train$Survived)
clean_train$Pclass <- as.factor(clean_train$Pclass)
clean_train$Sex <- as.factor(clean_train$Sex)
clean_train$Embarked <- as.factor(clean_train$Embarked)
clean_train$Ticket <- as.factor(clean_train$Ticket)
clean_train$Title <- as.factor(clean_train$Title)
clean_train$Surname <- as.factor(clean_train$Surname)
clean_train$FsizeD <- as.factor(clean_train$FsizeD)
clean_train$Deck <- as.factor(clean_train$Deck)


clean_test$Survived <- as.factor(clean_test$Survived)
clean_test$Pclass <- as.factor(clean_test$Pclass)
clean_test$Sex <- as.factor(clean_test$Sex)
clean_test$Embarked <- as.factor(clean_test$Embarked)
clean_test$Ticket <- as.factor(clean_test$Ticket)
clean_test$Title <- as.factor(clean_test$Title)
clean_test$Surname <- as.factor(clean_test$Surname)
clean_test$FsizeD <- as.factor(clean_test$FsizeD)
clean_test$Deck <- as.factor(clean_test$Deck)


categoricalResultCountBarchart <- function(data, column, categoryColumn) {
  survivors <- plyr::count(data, vars=c(column, categoryColumn))
  survivors <- group_by_(survivors, column) %>% dplyr::mutate(Percentage = round(freq * 100 / sum(freq))

  g <- ggplot(data = survivors, aes_string(x = column, y = "Percentage", fill = categoryColumn)) +
    geom_bar(stat="identity", position = "dodge") +
    geom_text(aes(label=sprintf("%d\n(%d %%)", freq, Percentage)))
  return (g)
}
```

4.1. Selección de los grupos de datos que se quieren analizar/comparar

Para este apartado solamente se consideran los datos del conjunto de entrenamiento.

```
# Inspeccionamos los datos de entrenamiento.
str(clean_train)
```

```
## 'data.frame':    891 obs. of  14 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 ...
##  $ Sex        : Factor w/ 2 levels "F","M": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 70.5 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     : Factor w/ 681 levels "110152","110413",..: 525 596 662 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  $ Title      : Factor w/ 5 levels "Master","Miss",..: 3 4 2 4 3 3 3 1 4 4 ...
##  $ Surname    : Factor w/ 667 levels "Abbing","Abbott",..: 74 137 256 203 12 414 383 468 297 431 ...
##  $ FsizeD     : Factor w/ 3 levels "large","singleton",..: 3 3 2 3 2 2 2 1 3 3 ...
##  $ Deck       : Factor w/ 11 levels "A","B","C","D",..: 11 3 11 3 11 11 5 11 11 10 ...
```

```
# Mostramos en forma de tabla
column_classes <- sapply(clean_train, class)
data <- data.frame(Variables = names(column_classes), Clases=unname(column_classes))
kable(data) %>%
    kable_styling(bootstrap_options = "striped", full_width = F)
```

Variables	Clases
PassengerId	integer
Survived	factor
Pclass	factor
Sex	factor
Age	numeric
SibSp	integer
Parch	integer
Ticket	factor
Fare	numeric
Embarked	factor
Title	factor
Surname	factor
FsizeD	factor
Deck	factor

De las características del conjunto de entrenamiento nos interesa analizar las variables cuantitativas `Age` y `Fare`;y las variables cuantitativas `Sex`, `Pclass`, `Title`, `FSizeD` y `Deck`. En principio descartaremos las variables cuantitativas `SibSp` y `Parch` debido a que están discretizadas en la variable cuantitativa `FSizeD`; y también las variables `Ticket` y `Surname` debido a que tienen demasiados valores.

Para analizar estas variables emplearemos diagramas de histogramas para las variables cuantitativas y diagramas de barras para las variables cualitativas en función de la supervivencia.

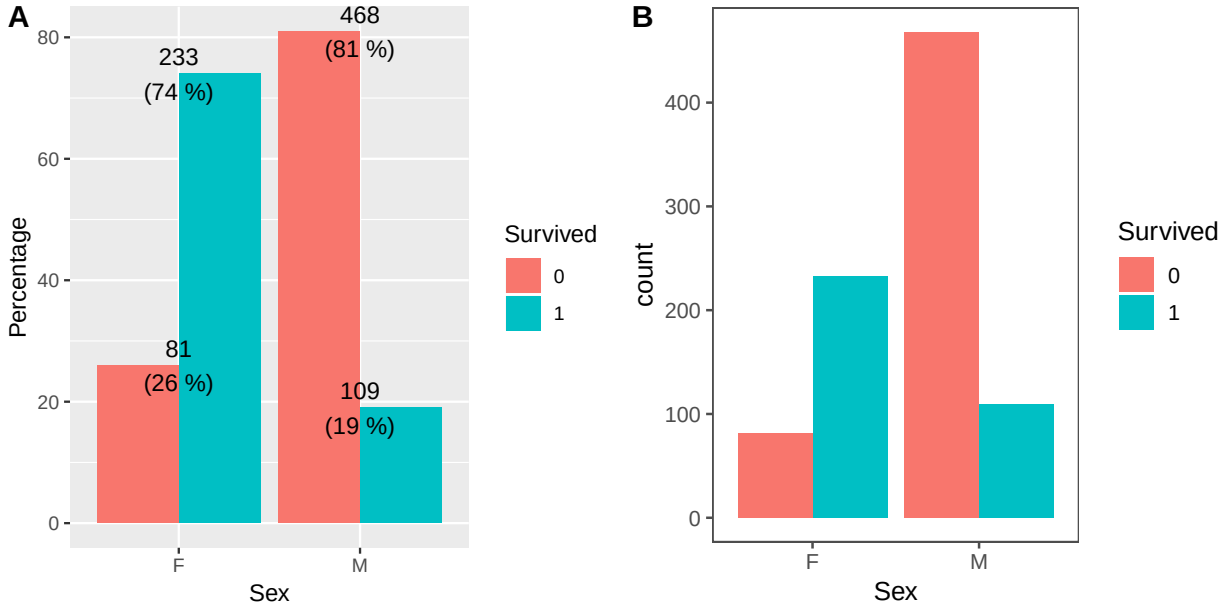
4.1.1 Relaciones de Características con la Supervivencia

Sexo (Sex) y Supervivencia (Survived)

```
# Porcentaje de supervivencia por Sexo
plot.sex.per <-  categoricalResultCountBarchart(clean_train, "Sex", "Survived")
```

```
## Warning: group_by_() is deprecated.
## Please use group_by() instead
##
## The 'programming' vignette or the tidyeval book can help you
## to program with group_by() : https://tidyeval.tidyverse.org
## This warning is displayed once per session.
```

```
plot.sex <- ggplot(clean_train, aes(x = Sex, fill = factor(Survived))) +
    geom_bar(position=position_dodge()) +
    labs(fill = "Survived") +
    theme_few()

#ggarrange(plot.sex.per, plot.sex)
ggarrange(plot.sex.per, plot.sex, labels = c("A", "B"), ncol = 2, nrow = 1)
```



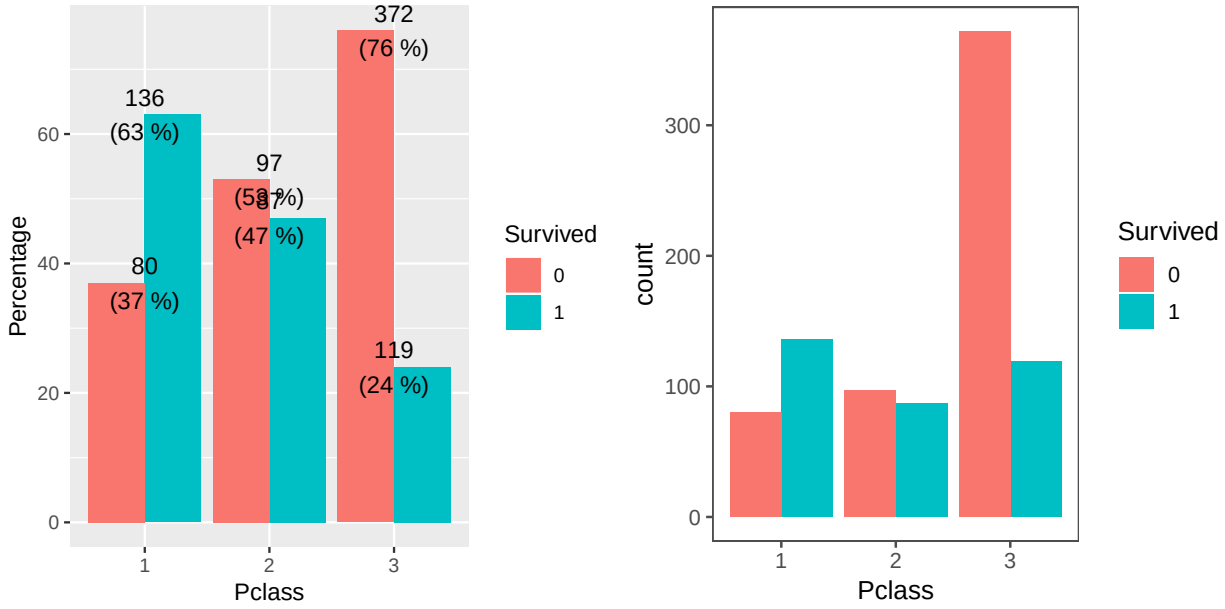
El diagrama de barras anterior muestra la distribución de supervivencia de mujeres y hombres. Como se intuía está característica parece influir en la supervivencia. El gráfico de barras muestra que un 74% de los *pasajeros mujeres* sobrevivieron, mientras que solo un 19% de los *pasajeros varones* sobrevivieron.

De tal forma que aquellos pasajeros con sexo femenino tuvieron una tasa de supervivencia más alta que los varones.

Clase (Pclass) y Supervivencia (Survived)

Otra característica que puede influir en la supervivencia es la condición socioeconómica. Esta condición se expresa a través de la variable Pclass (Clase del pasajero).

```
plot.Pclass.per <- categoricalResultCountBarchart(clean_train, "Pclass", "Survived")

# Mostramos la relacion entre el Pclass y la Supervivencia
plot.Pclass <- ggplot(clean_train, aes(Pclass, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  labs(fill = "Survived") +
  theme_few()

ggarrange(plot.Pclass.per, plot.Pclass, ncol = 2, nrow = 1)
```



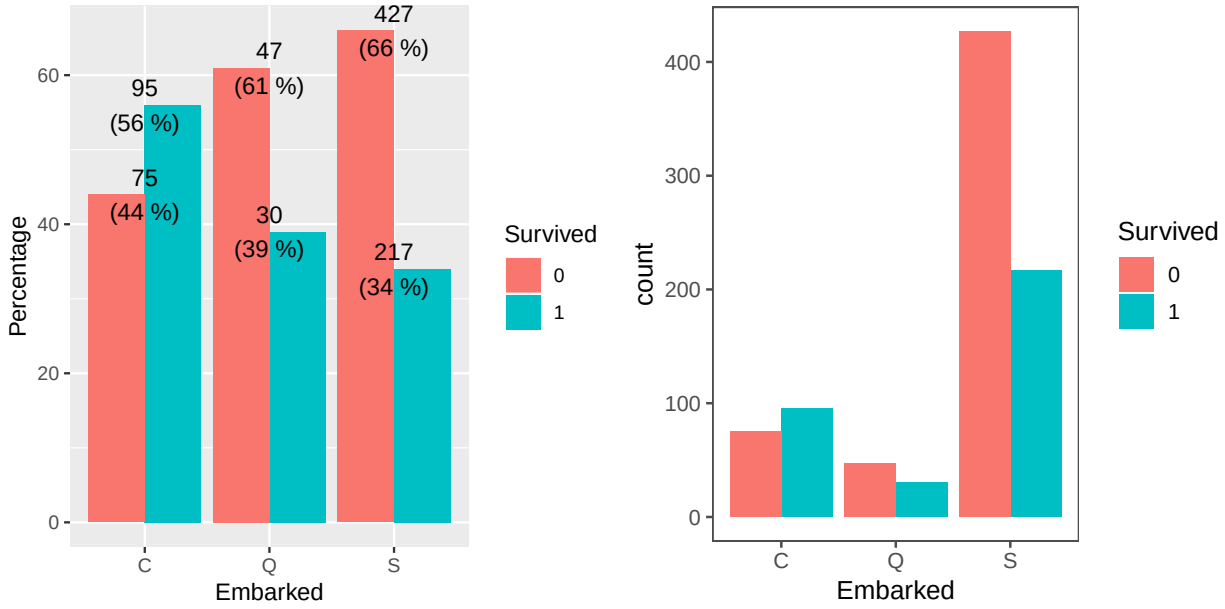
Las gráficas anteriores muestran la distribución de la supervivencia en función de la clase del pasajero. En el gráfico se observa que esta característica parece influir en la supervivencia. El gráfico de barras muestra que sobre el 63 % de los pasajeros de *primera clase* sobrevivieron, mientras que sobre el 48 % de los pasajeros de *segunda clase* sobrevivieron, y solo el 24 % de los pasajeros de *tercera clase* sobrevivieron.

De tal forma que aquellos pasajeros en las clases más altas tienen una tasa de supervivencia más alta que aquellos pasajeros en las clases más bajas.

Embarque (Embarked) y Supervivencia (Survived)

Otra característica que se desea analizar es la influencia del embarque en la supervivencia.

```
# Mostramos la relacion entre el Embarque y la Supervivencia
plot.Embarked.per <- categoricalResultCountBarchart(clean_train, "Embarked", "Survived")

# Mostramos la relacion entre el Embarque y la Supervivencia
plot.Embarked <- ggplot(clean_train, aes(Embarked, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  labs(fill = "Survived") +
  theme_few()

ggarrange(plot.Embarked.per, plot.Embarked, ncol = 2, nrow = 1)
```

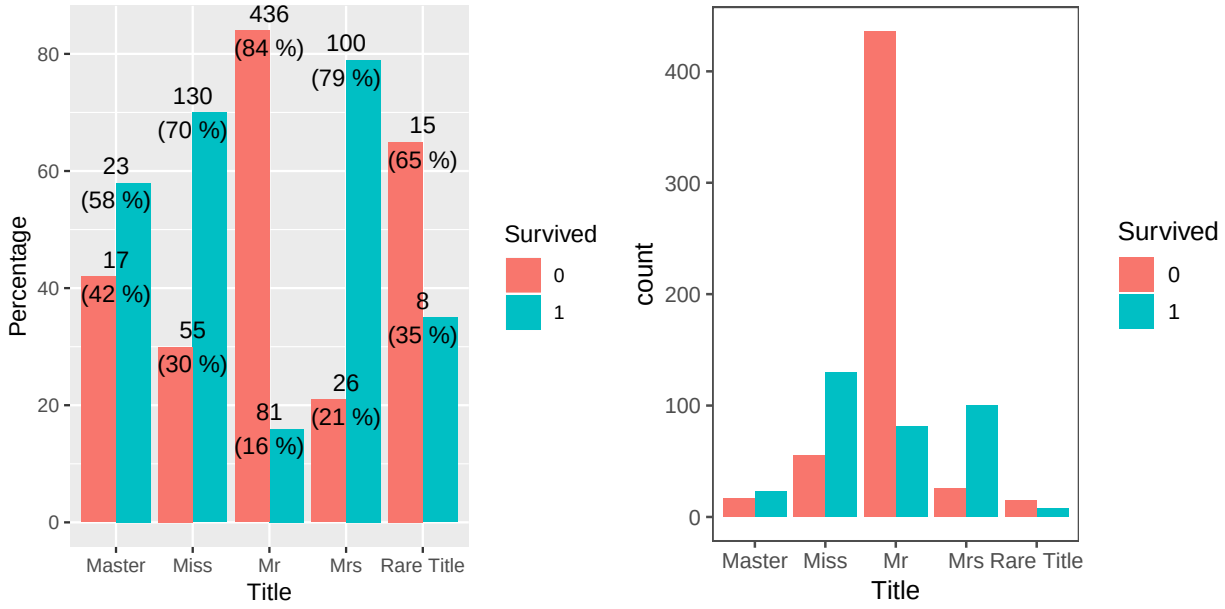


La gráfica anterior muestra que la mayoría de los pasajeros parece ser que embarcaron en Southampton (S). Por otra parte, más del 60% de los pasajeros que embarcaron en Southampton (S) murieron. Mientras, más del 60% de los pasajeros que embarcaron en Cherburgo (C) sobrevivieron.

Titulo (Title) y Supervivencia (Survived)

Otra característica que se desea analizar es la influencia del Título en la supervivencia. Estos suelen estar asociados al sexo del pasajero y su estado social. Suponemos que los hombres deberían tener una tasa de mortalidad más alta debido a que tienen menos prioridad en el momento de embarcar en un bote salvavidas.

```
# Mostramos la relacion entre el Titulo (Title) y Supervivencia (Survived)
plot.Title.per <- categoricalResultCountBarchart(clean_train, "Title", "Survived")

plot.Title <- ggplot(clean_train, aes(Title, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  labs(fill = "Survived") +
  theme_few()

ggarrange(plot.Title.per, plot.Title, ncol = 2, nrow = 1)
```

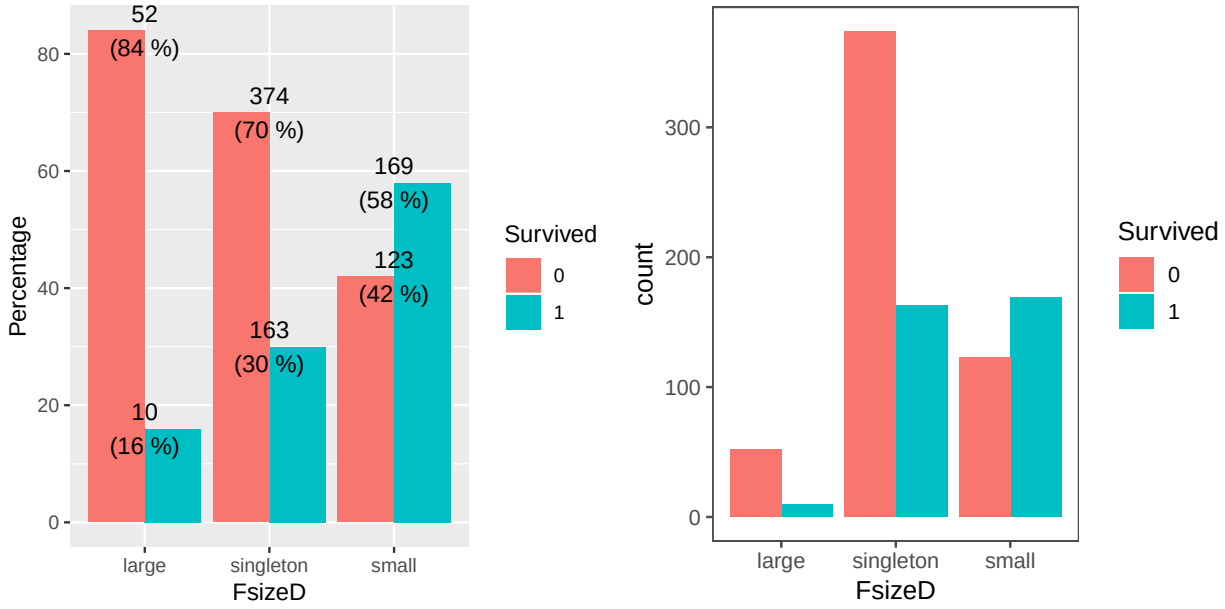


La gráfica anterior parece confirmar nuestra suposición que los pasajeros con título Mr (varones) solo el 16% sobrevivió.

Tamaño de la Familia (FsizeD) y Supervivencia (Survived)

Otra característica que se desea analizar es la influencia del Tamaño de la familia en la supervivencia.

```
# Mostramos la relacion entre la Clase (FsizeD) y Supervivencia (Survived)
plot.FsizeD.per <- categoricalResultCountBarchart(clean_train, "FsizeD", "Survived")

plot.FsizeD <- ggplot(clean_train, aes(FsizeD, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  labs(fill = "Survived") +
  theme_few()

ggarrange(plot.FsizeD.per, plot.FsizeD, ncol = 2, nrow = 1)
```



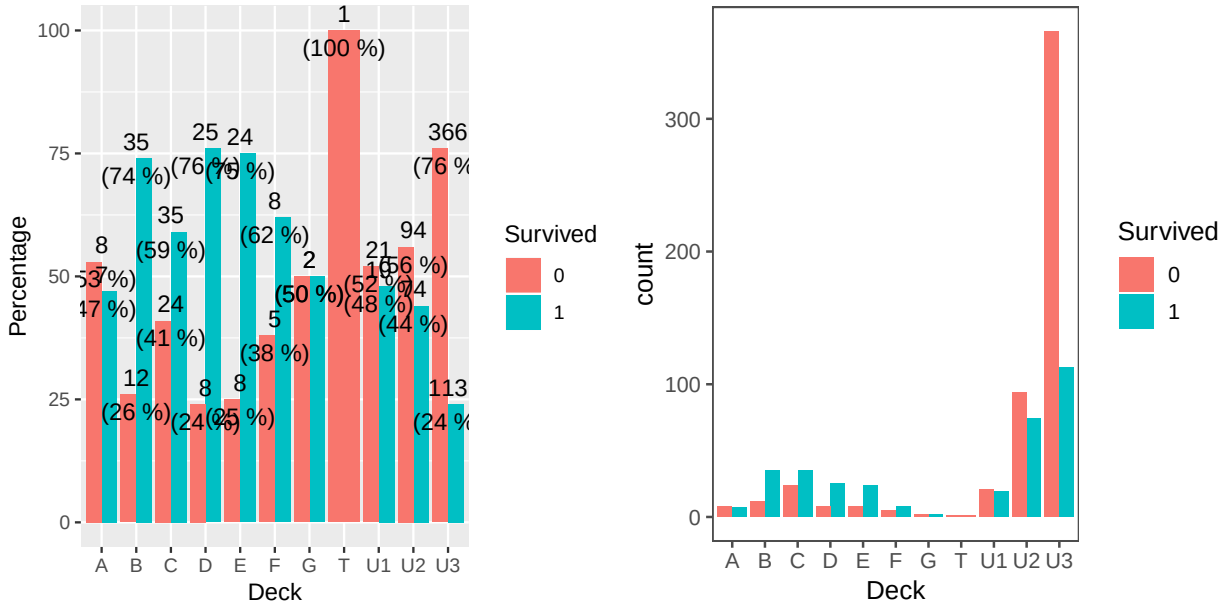
La gráfica anterior muestra que sobre el 70% los pasajeros solteros y sobre el 82% de las familias grandes no sobrevivieron. Respecto al conjunto de solteros suponemos que la mayoría deberían ser varones dado que en la época del accidente sería más habitual que estos viajen solos. Además, suponemos que las familias grandes no cabrían todos en un bote de salvavidas y esto podría influir en su supervivencia.

Más adelante analizaremos esta característica en función del sexo ya que ser soltero y varón debería ser un rasgo que influya en la supervivencia.

Clase (Deck) y Supervivencia (Survived)

La característica de la cubierta (Deck) esta relacionada con la condición socioeconómica (Clase) y la ubicación de su camarote en el barco. De tal forma que pasajeros más cercanos de la cubierta de A están más cerca de los botes salvavidas. Además, durante la imputación de valores perdidos se asignado el valor U1 a los pasajeros de primera clase y probablemente estarán más cerca de los botes salvavidas. Por tanto, esta característica puede influir en la supervivencia.

```
plot.Deck.per <- categoricalResultCountBarchart(clean_train, "Deck", "Survived")

plot.Deck <- ggplot(clean_train, aes(Deck, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  labs(fill = "Survived") +
  theme_few()

ggarrange(plot.Deck.per, plot.Deck, ncol = 2, nrow = 1)
```



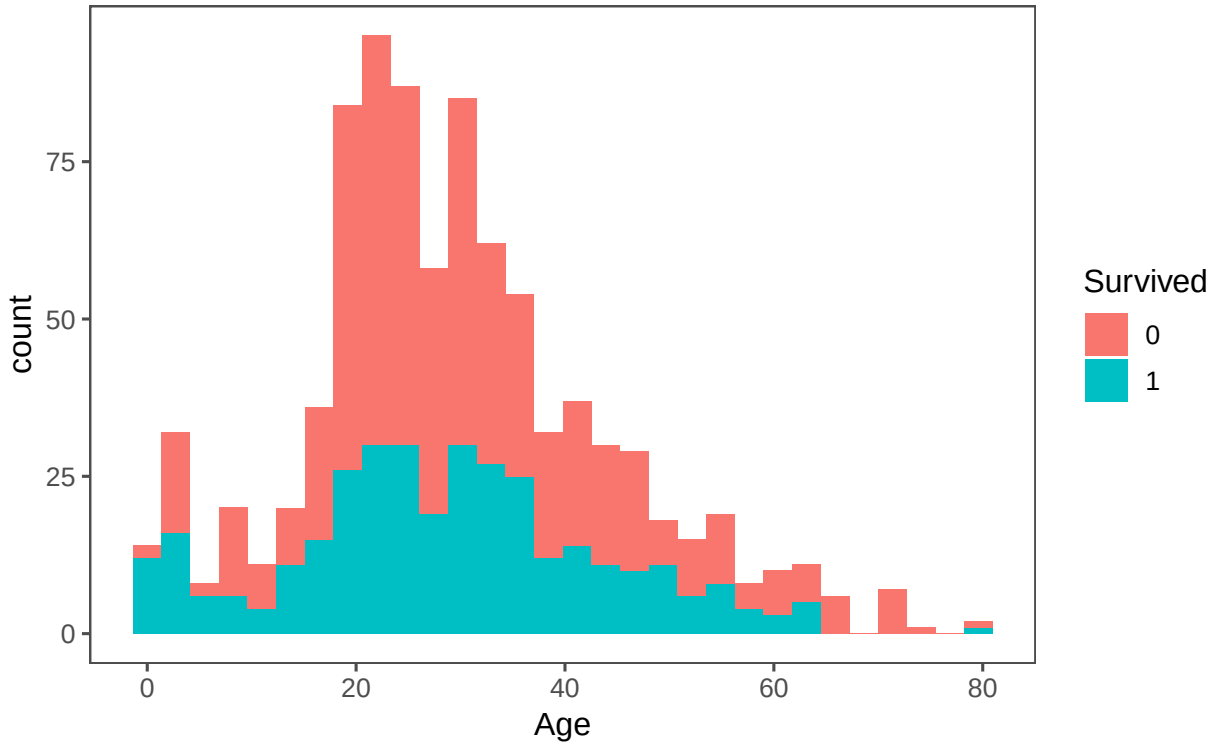
La gráfica anterior muestra que muchos pasajeros asignados a cubiertas con una clase baja no sobrevivieron posiblemente porque están en cubiertas más alejadas a los botes.

Edad (Age) y Supervivencia (Survived)

Otra característica que puede influir en la supervivencia es la edad debido a que los accidentes los menores de edad deberían tener preferencia a la hora de emplear los botes salvavidas.

```
# Mostramos
ggplot(clean_train, aes(Age, fill = factor(Survived))) +
  geom_histogram() +
  labs(fill = "Survived") +
  theme_few()
```

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



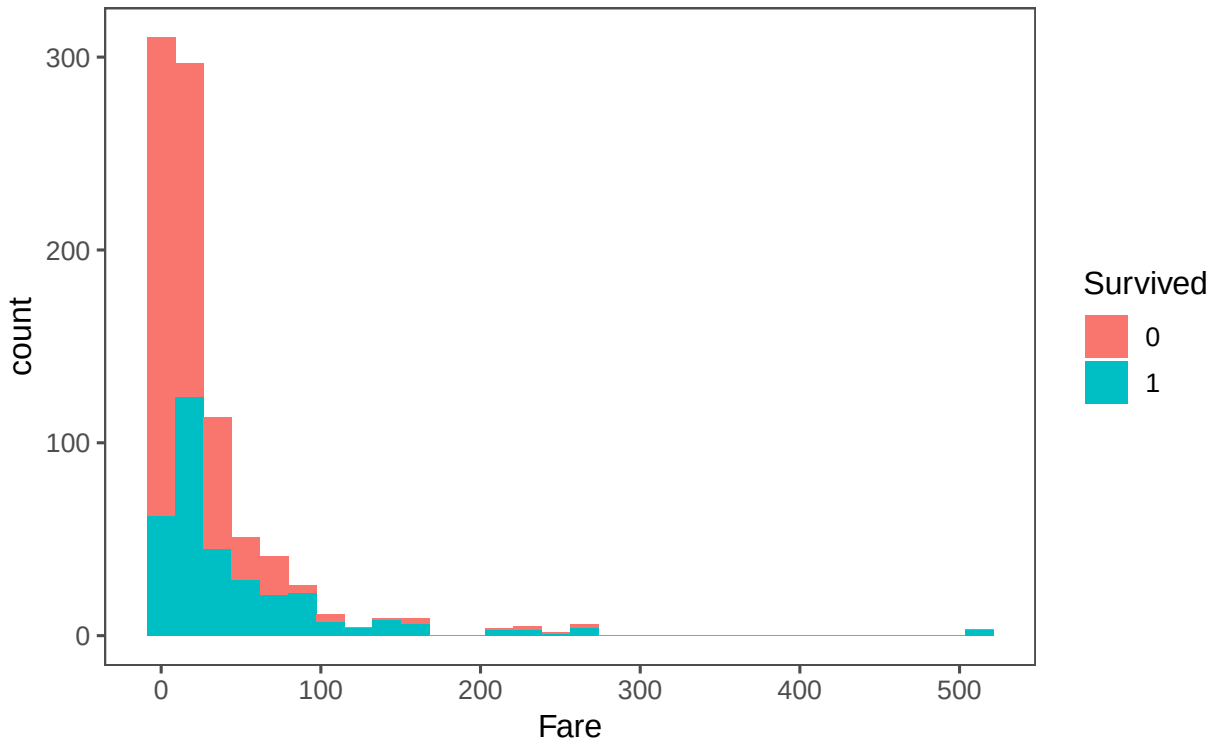
No hay nada fuera de lo común en esta trama, excepto la parte izquierda de la distribución. Demuestra que los niños y los bebés eran la prioridad, por lo tanto, se salvó una buena parte de los bebés.

Tarifa (Fare) y Supervivencia (Survived)

Otra característica que también puede influir en la supervivencia es la tarifa que puede estar relacionada con la condición socioeconómica.

```
# Mostramos
ggplot(clean_train, aes(Fare, fill = factor(Survived))) +
  geom_histogram() +
  labs(fill = "Survived") +
  theme_few()
```

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



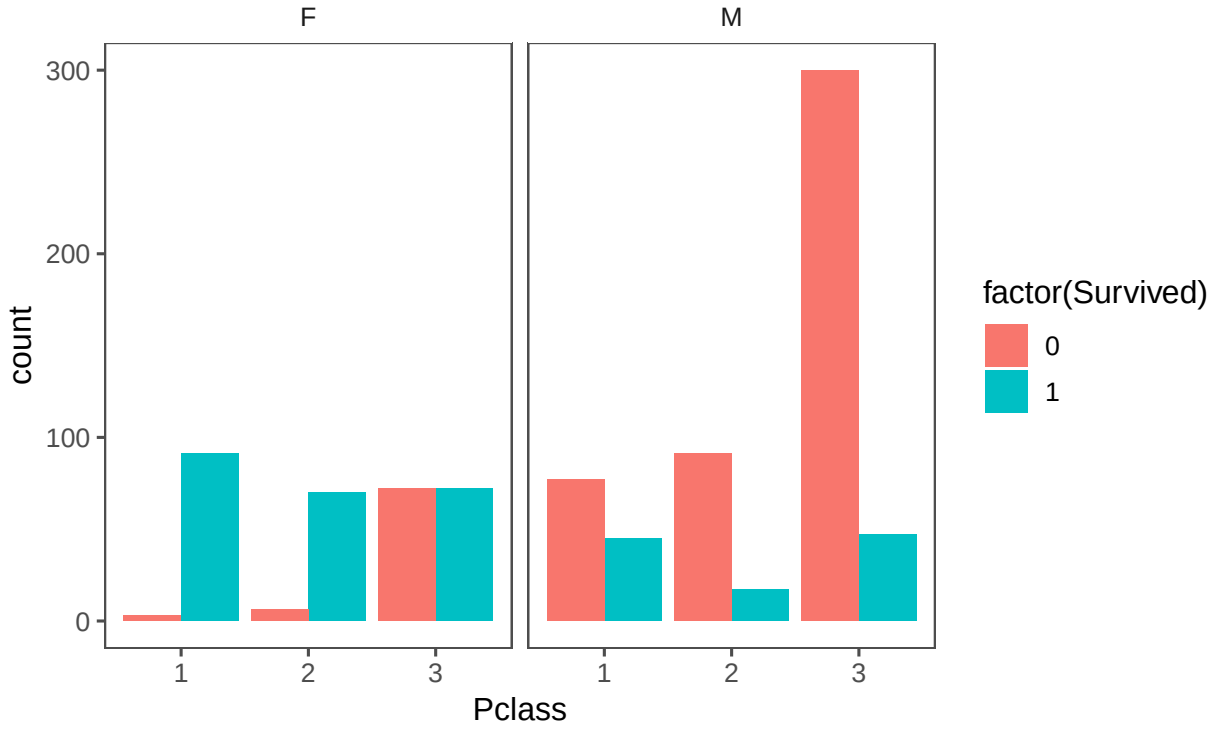
La gráfica anterior muestra algo interesante, existe un pico en los valores de menos de 100 dólares que representa que muchos de los pasajeros que compraron un ticket dentro de ese rango no sobrevivieron. Cuando la tarifa es aproximadamente más de 280 dólares, la tasa de mortalidad es baja, lo que significa que todos los que pasaron de esa la tarifa sobrevivieron.

4.1.2 Relaciones de características combinadas con la Supervivencia.

En esta sección, vamos a analizar más de dos relaciones de características en un solo gráfico.

Supevivencia por Clase (Pclass) y Sexo (Sex)

```
# Mostramos la relacion entre el Clase (Pclass) y la Supervivencia según el Sexo (Sex) del pasajero
ggplot(clean_train, aes(Pclass, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  facet_grid(.~Sex) +
  theme_few()
```

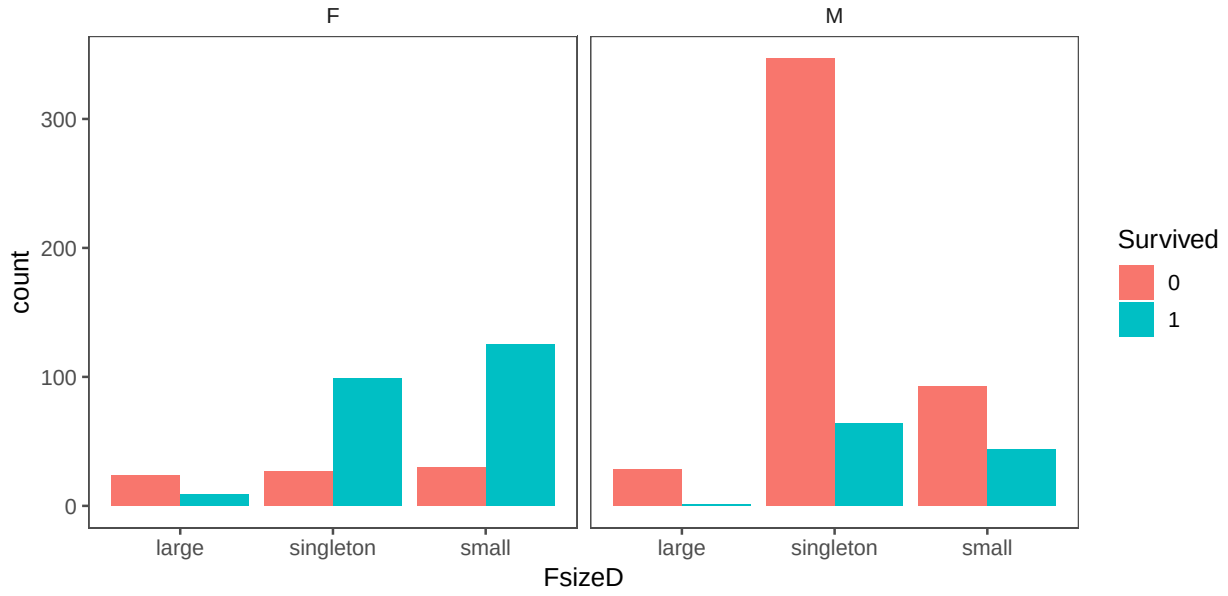


La gráfica anterior muestra que los pasajeros mujeres y de una clase alta sobrevivieron en su mayoría. También se observa que los pasajeros varones tuvieron una tasa de supervivencia mucho más baja que las mujeres. Esta tasa de supervivencia va empeorando a medida que la clase del pasajero baje.

Se puede concluir que ser de sexo y la clase pueden influir en la supervivencia del pasajero.

Supevivencia por Tamaño de Familia (FsizeD) y Sexo (Sex)

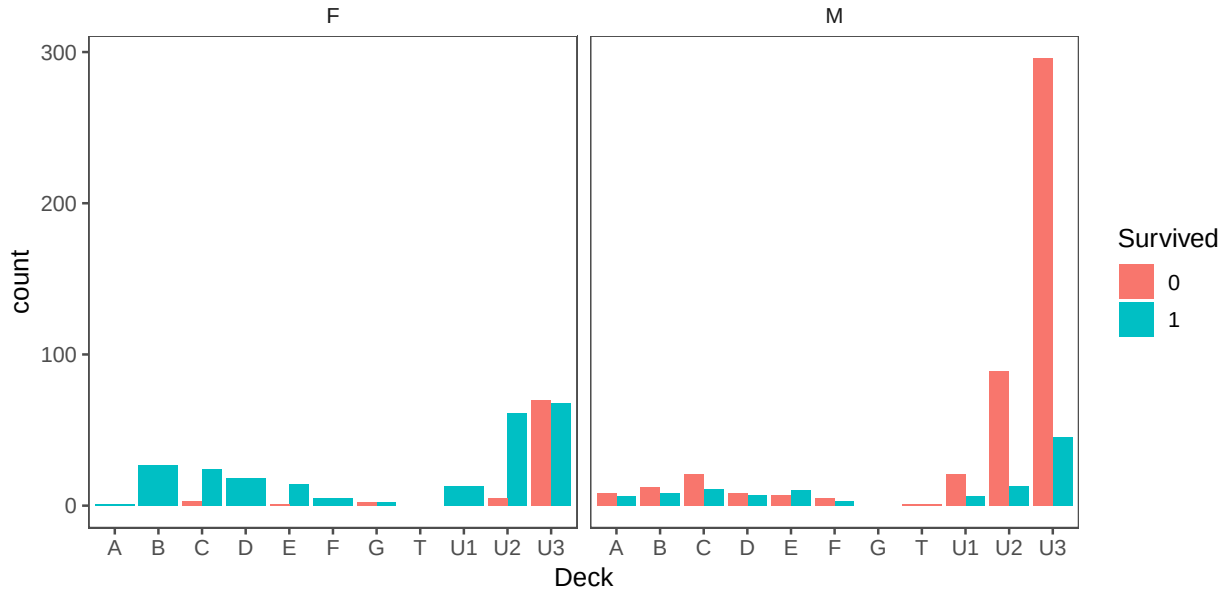
```
# Mostramos la relacion entre el FsizeD y la Supervivencia según el Sexo del pasajero
ggplot(clean_train, aes(FsizeD, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  labs(fill = "Survived") +
  facet_grid(.~Sex) +
  theme_few()
```



La gráfica anterior muestra que los pasajeros solteros varones tuvieron una tasa de mortalidad más alta. Esta valor es lógico debido que en los botes salvavidas tendrían una preferencia menor a las mujeres y niños.

Supevivencia por Cubierta (Deck) y Sexo (Sex)

```
# Mostramos la relacion entre el Deck y la Supervivencia según el Sexo del pasajero
ggplot(clean_train, aes(Deck, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  labs(fill = "Survived") +
  facet_grid(.~Sex) +
  theme_few()
```



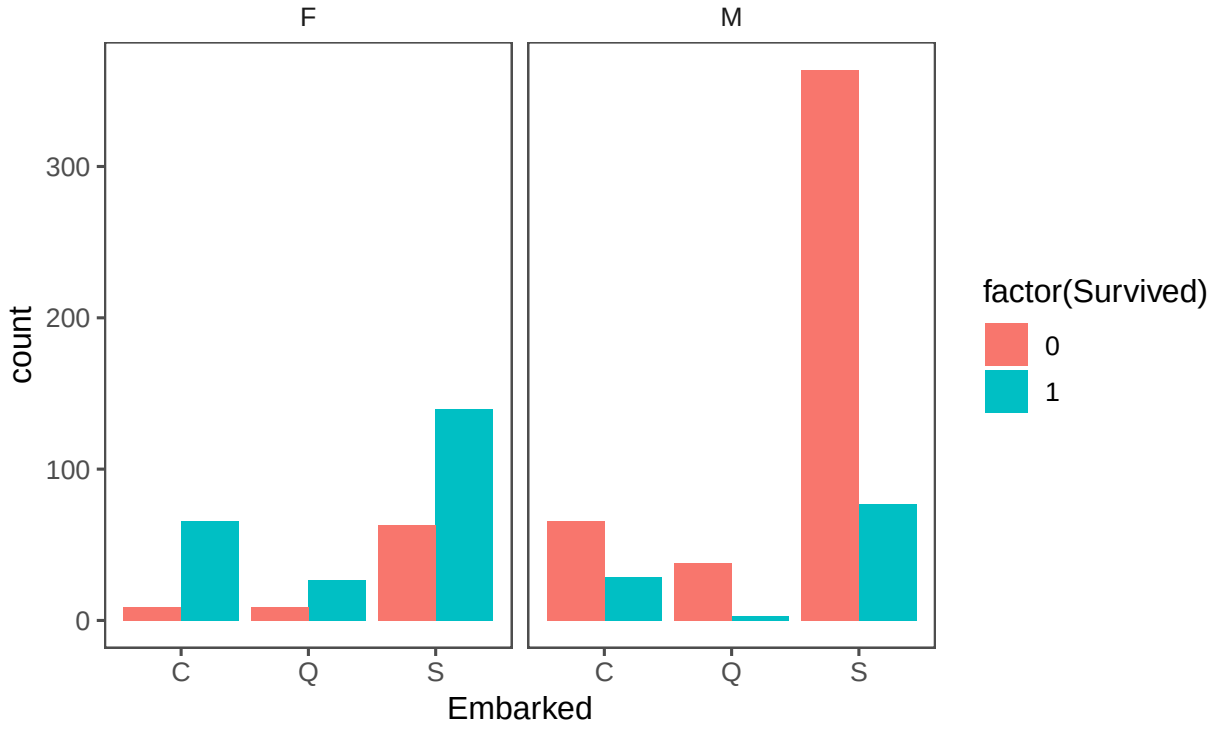
La gráfica anterior muestra que las probabilidades de supervivencia aumenten si era mujer. Mientras que para los pasajeros varones su supervivencia disminuye a medida que su camarote está más alejado de la

cubierta principal donde están los botes salvavidas.

¿La supervivencia que afecta a los varones se puede deber a que estaban muy alejados de los botes o se debido a la clase del pasajero?

Supevivencia por Embarque (Embarked) y Sexo (Sex)

```
# Mostramos la relacion entre el Embarque y la Supervivencia según el sexo del pasajero
ggplot(clean_train, aes(Embarked, fill = factor(Survived))) +
  geom_bar(position=position_dodge()) +
  facet_grid(.~Sex) +
  theme_few()
```

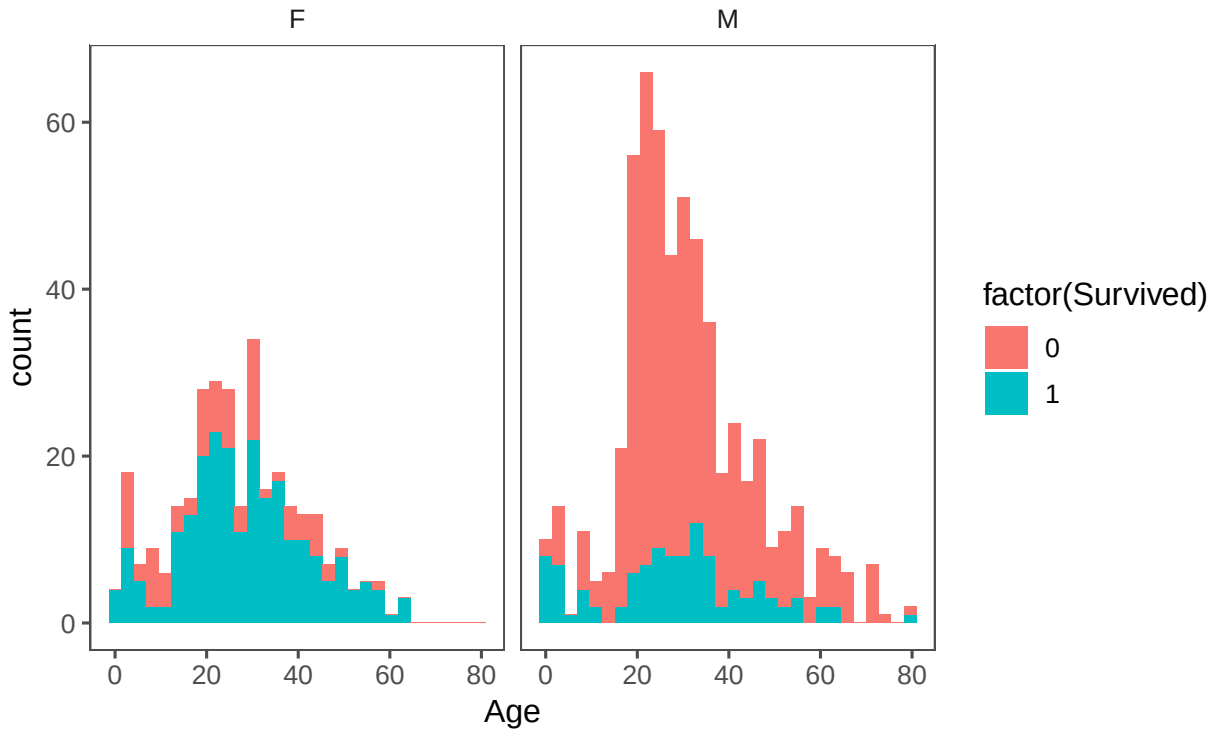


La gráfica anterior muestra que la mayoría de los pasajeros parece ser que embarcaron en Southampton (S). Aunque la mayoría de los pasajeros embarco en Southampton (S) a priori no debería ser relevante para la supervivencia, a menos que tenga alguna relación con la localización del camarote o la clase del pasajero.

Supevivencia por Edad (Age) y Sexo (Sex)

```
# Mostramos la Supevivencia por Edad (Age) y Sexo (Sex)
ggplot(clean_train, aes(x = Age, fill = factor(Survived))) +
    geom_histogram() +
    facet_grid(.~Sex) +
    theme_few()
```

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



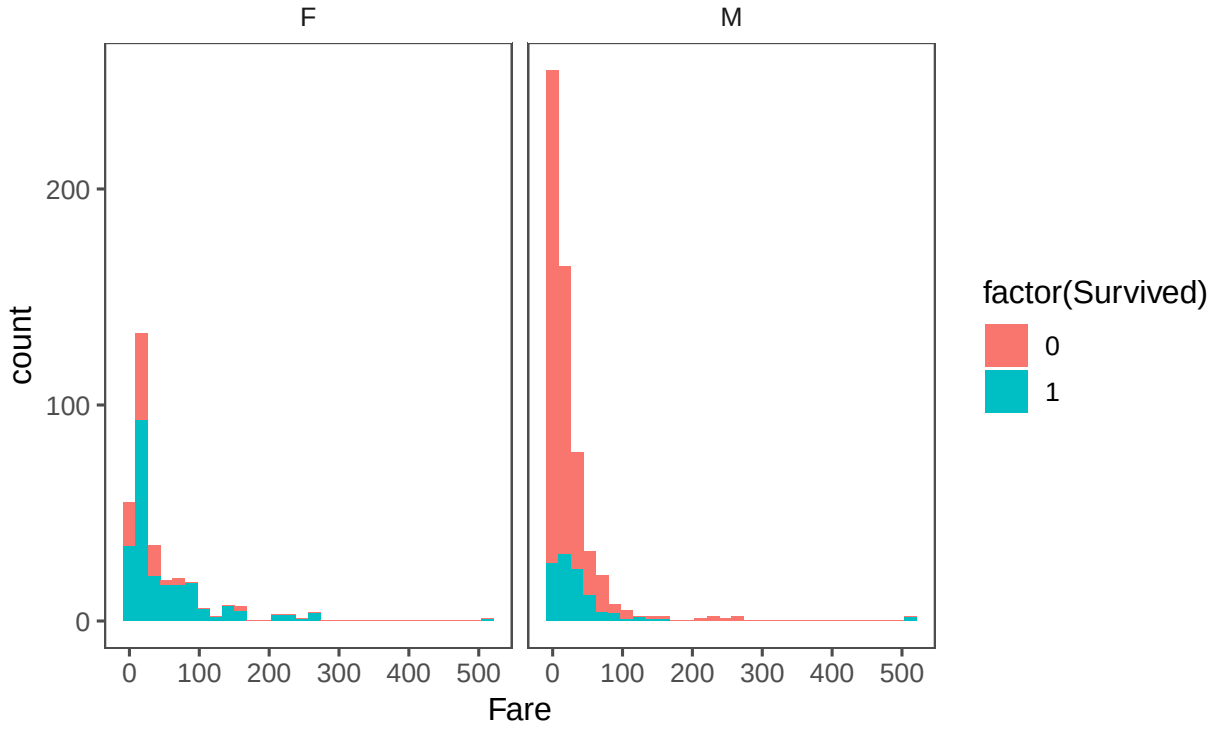
La gráfica anterior muestra que la supervivencia de los varones es baja para los adultos. Los niños varones tienen una tasa de supervivencia alta, esto es lógico debido a la preferencia que tuvieron estos en los botes.

Por tanto, podemos concluir que el sexo y la edad de los pasajeros son características que influyen en la supervivencia.

Supevivencia por Tarifa (Fare) y Sexo (Sex)

```
# Mostramos la Supevivencia por Tarifa (Fare) y Sexo (Sex)
ggplot(clean_train, aes(x = Fare, fill = factor(Survived))) +
    geom_histogram() +
    facet_grid(.~Sex) +
    theme_few()
```

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



La gráfica anterior no se observa algo nuevo. Solamente que la condición socioeconómica parece un factor que puede influir en la supervivencia.

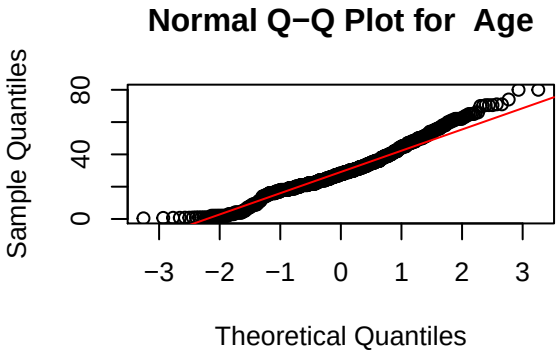
De las gráficas anteriores se concluye que las características `Age`,`Sex`, `Fare` y `Pclass` parecen tener influyen en la supervivencia.

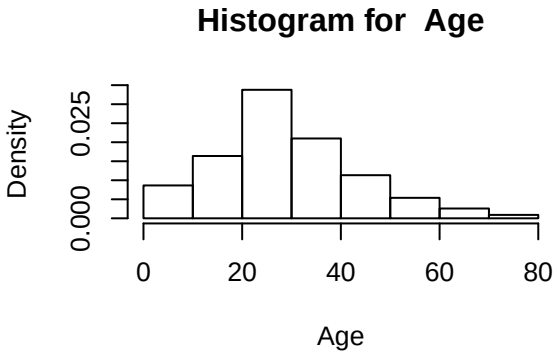
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

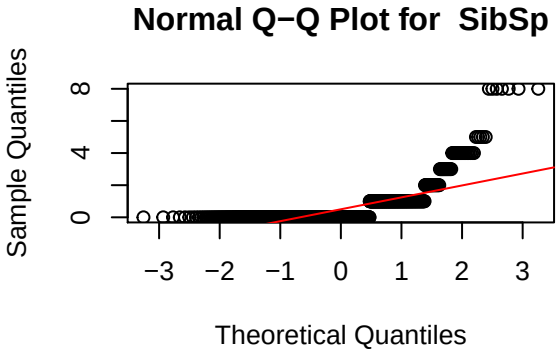
4.4.2. Normalidad

Para revisar si las variables pueden ser candidatas a la normalización miramos las gráficas de quantile-quantile plot y el histograma.

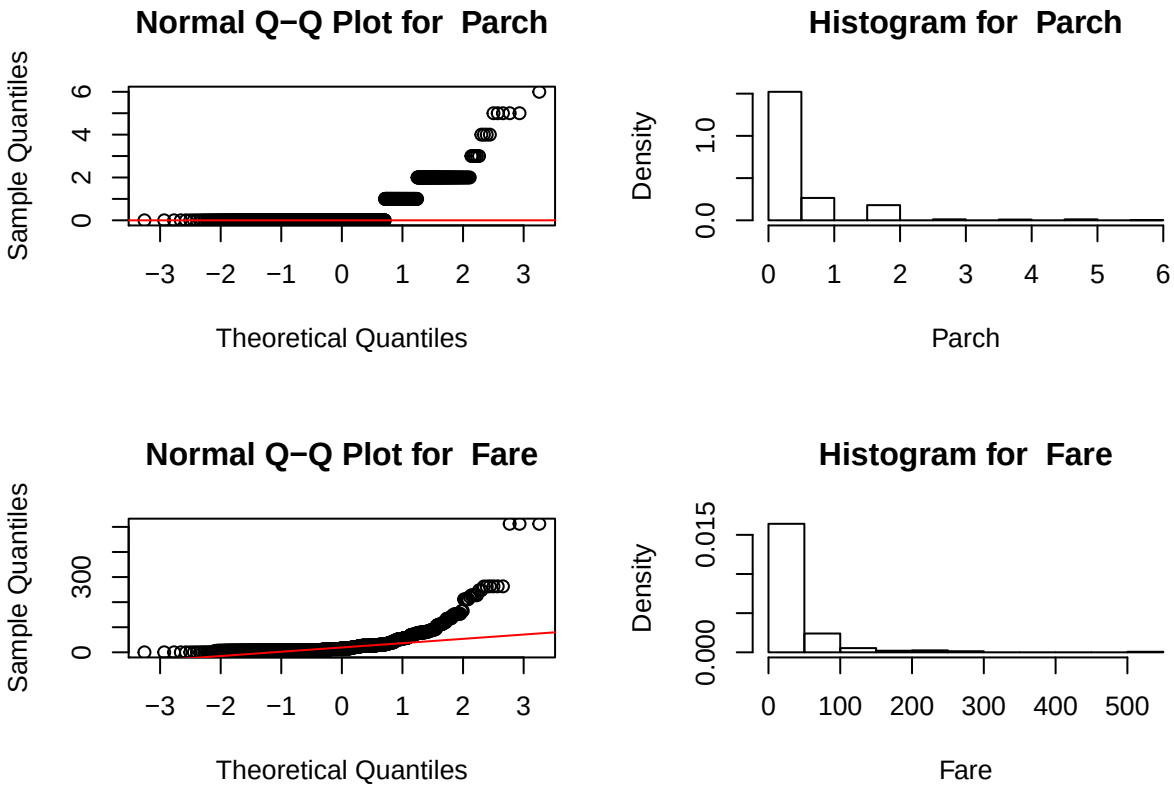
```
alpha = 0.05
drawQQPlotAndtHist <- function(dataset) {
  par(mfrow=c(2,2))
  for(i in 1:ncol(dataset)) {
    if (is.numeric(dataset[,i])){
      qqnorm(dataset[,i],main = paste("Normal Q-Q Plot for ",colnames(dataset)[i]))
      qqline(dataset[,i],col="red")
      hist(dataset[,i],
        main=paste("Histogram for ", colnames(dataset)[i]),
      xlab=colnames(dataset)[i], freq = FALSE)
    }
  }
}
# Mostramos las gráficas.
dataset <- select(clean_train, -PassengerId)
drawQQPlotAndtHist(dataset)
```


Normal Q−Q Plot for Age
Sample Quantiles
0
40
80
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Histogram for Age
Density
0.000
0.025
0
20
40
60
80
Age


Normal Q−Q Plot for SibSp
Sample Quantiles
0
4
8
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Histogram for SibSp
Density
0.0
0.6
0
2
4
6
8
SibSp



De las gráficas anteriores, se observa que la característica **Age** pueden ser candidata a la normalización. No obstante, se aplicará el test de Shapiro-Wilk para constratar esta asunción.

Test Shapiro-Wilk

El test de Shapiro-Wilk se usa para contrastar si un conjunto de datos siguen una distribución normal o no. En nuestro caso se aplicará este test cada una las variables cuantitativas consideradas.

De tal forma que la hipótesis nula (H_0) y la alternativa (H_1) se pueden escribir de la siguiente forma:

Hipótesis nula (H_o): Los datos de la muestra *no son significativamente diferentes* de una población normal.

Hipótesis alternativa (H_1): Los datos de la muestra *son significativamente diferentes* de una población normal.

Zona de rechazo. Para todo valor de probabilidad mayor que un nivel de significación $\alpha = 0.05$, se acepta H_o y se rechaza H_1.

Para comprobar la asunción de normalidad aplicamos el test Shapiro-Wilk, para ello utilizamos la función **`shapiro.test`**. A continuación, se muestra la aplicación del test Shapiro-Wilk para las variables cuantitativas consideradas:

```
# Test Shapirp para Age
shapiro.test(clean_train$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  clean_train$Age
## W = 0.97565, p-value = 4.796e-11
```

```r
# Test Shapirp para SibSp
shapiro.test(clean_train$SibSp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  clean_train$SibSp
## W = 0.51297, p-value < 2.2e-16
```

```r
# Test Shapirp para Parch
shapiro.test(clean_train$Parch)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  clean_train$Parch
## W = 0.53281, p-value < 2.2e-16
```

```r
# Test Shapirp para Fare
shapiro.test(clean_train$Fare)
```

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

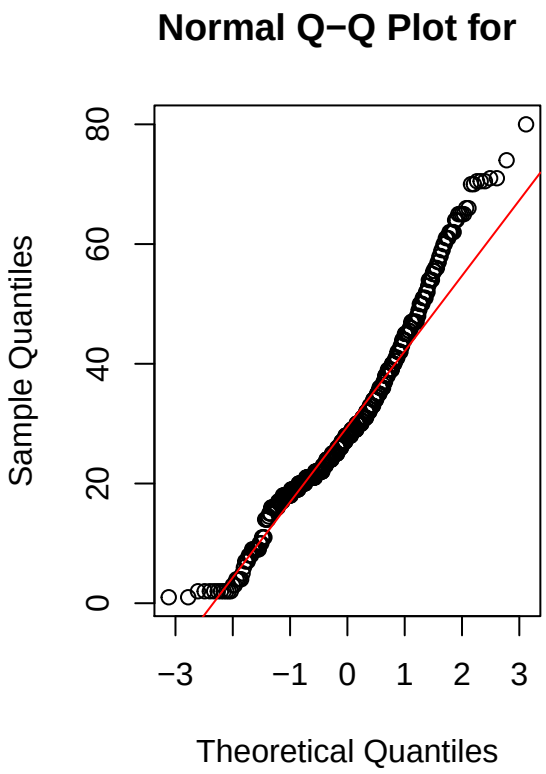
Dado los resultados anteriores, se observa que para las cuatro características consideradas sus correspondientes p-valores son inferiores al nivel de significación ($\alpha = 0.05$). Por tanto, rechazamos la H_0 y concluimos con un 95% de confianza que los datos no se distribuyen normalmente.

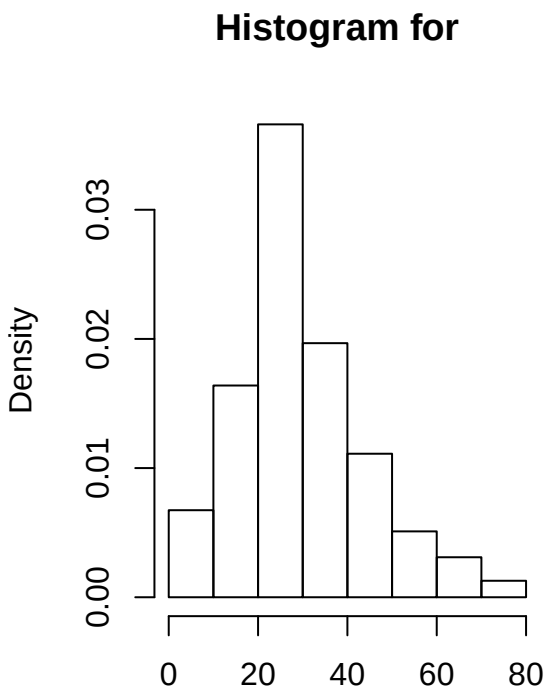
Normalidad de Supervivencia y Edad

Ahora se aplicará este test para realizar el contraste de si existen diferencias en la edad (Age) en función de la supervivencia (Survived).

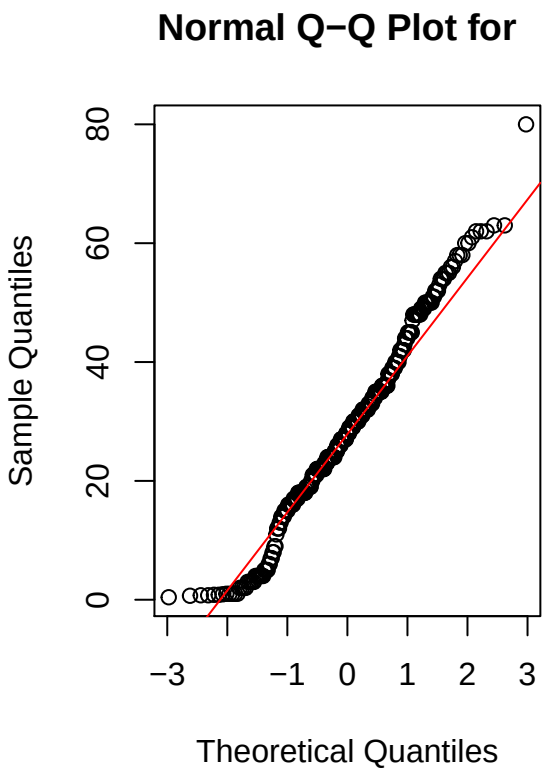
```r
age_sur_0 <- clean_train$Age[clean_train$Survived==0]
age_sur_1 <- clean_train$Age[clean_train$Survived==1]
```

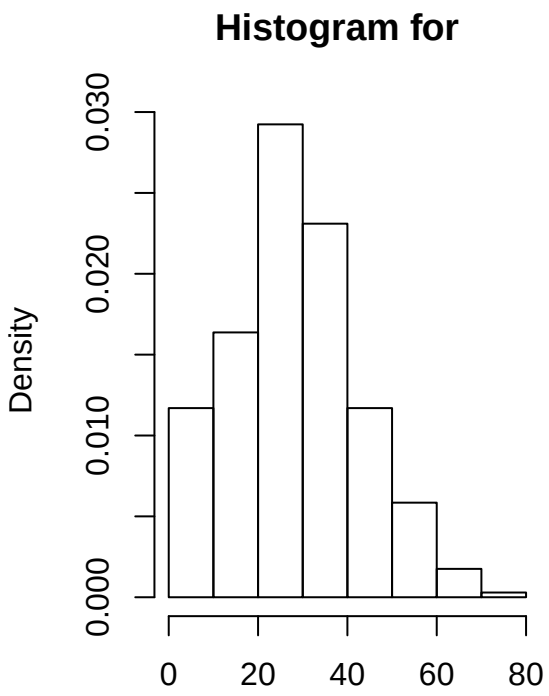
```r
par(mfrow=c(1,2))
qqnorm(age_sur_0, main = paste("Normal Q-Q Plot for ", colnames(age_sur_0)[1]))
qqline(age_sur_0, col="red")
hist(age_sur_0,
  main=paste("Histogram for ", colnames(age_sur_0)[1]),
xlab=colnames(age_sur_0)[1], freq = FALSE)
```






```
par(mfrow=c(1,2))
qqnorm(age_sur_1, main = paste("Normal Q-Q Plot for ", colnames(age_sur_1)[1]))
qqline(age_sur_1, col="red")
hist(age_sur_1,
  main=paste("Histogram for ", colnames(age_sur_1)[1]),
xlab=colnames(age_sur_1)[1], freq = FALSE)
```






```
# Test Shapirp para Fare
shapiro.test(age_sur_0)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_sur_0
## W = 0.96023, p-value = 5.094e-11
```

```
# Test Shapirp para Fare
shapiro.test(age_sur_1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  age_sur_1
## W = 0.98349, p-value = 0.0005847
```

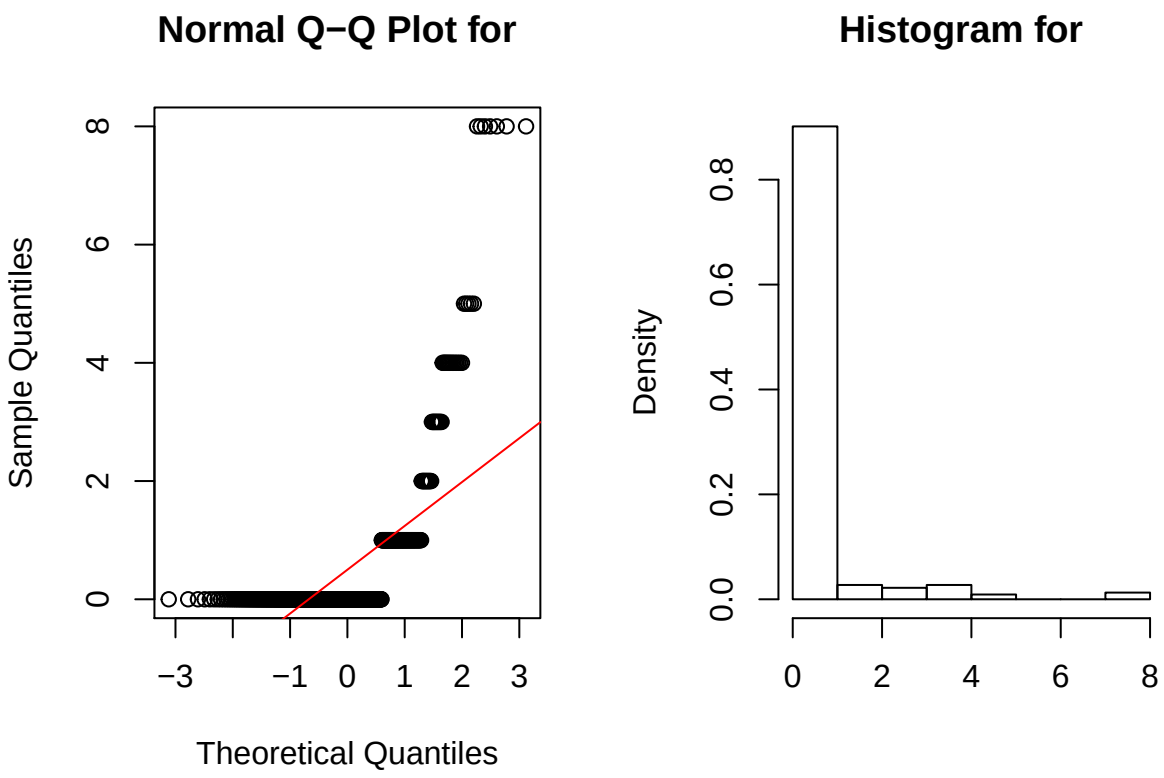
Dado los resultados anteriores, se observa que para las cuatro características consideradas sus correspondientes p-valores son inferiores al nivel de significación ($\alpha = 0.05$). Por tanto, rechazamos la H_0 y concluimos con un 95% de confianza que los datos no se distribuyen normalmente.

Normalidad de Supervivencia y Número de hermanos/cónyuges a bordo (SibSp)

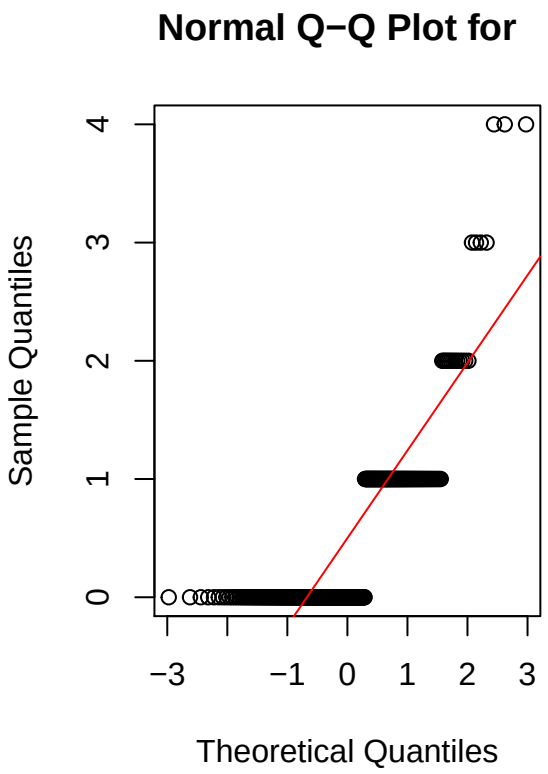
Ahora se aplicará este test para realizar el contraste de si existen diferencias en la característica Número de hermanos/cónyuges a bordo (SibSp) en función de la supervivencia (Survived).

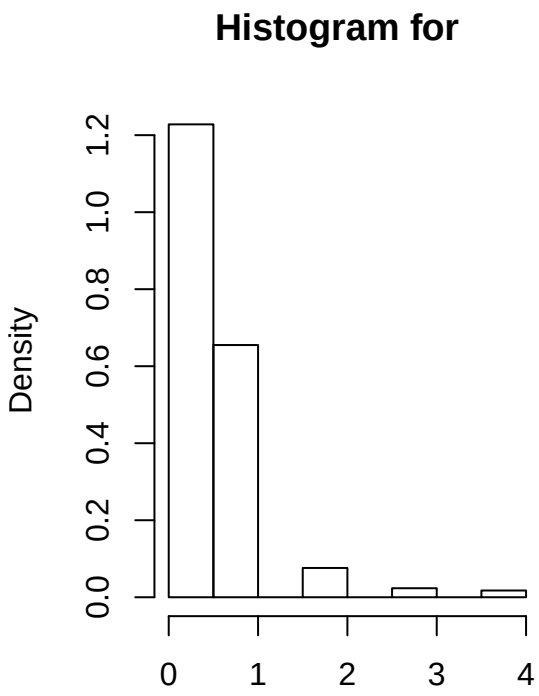
```
SibSp_sur_0 <- clean_train$SibSp[clean_train$Survived==0]
SibSp_sur_1 <- clean_train$SibSp[clean_train$Survived==1]
```

```
par(mfrow=c(1,2))
qqnorm(SibSp_sur_0, main = paste("Normal Q-Q Plot for ", colnames(SibSp_sur_0)[1]))
qqline(SibSp_sur_0, col="red")
hist(SibSp_sur_0,
  main=paste("Histogram for ", colnames(SibSp_sur_0)[1]),
xlab=colnames(SibSp_sur_0)[1], freq = FALSE)
```




```
par(mfrow=c(1,2))
qqnorm(SibSp_sur_1, main = paste("Normal Q-Q Plot for ", colnames(SibSp_sur_1)[1]))
qqline(SibSp_sur_1, col="red")
hist(SibSp_sur_1,
  main=paste("Histogram for ", colnames(SibSp_sur_1)[1]),
xlab=colnames(SibSp_sur_1)[1], freq = FALSE)
```






```
# Test Shapirp para Fare
shapiro.test(SibSp_sur_0)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  SibSp_sur_0
## W = 0.48418, p-value < 2.2e-16
```

```
# Test Shapirp para Fare
shapiro.test(SibSp_sur_1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  SibSp_sur_1
## W = 0.65477, p-value < 2.2e-16
```

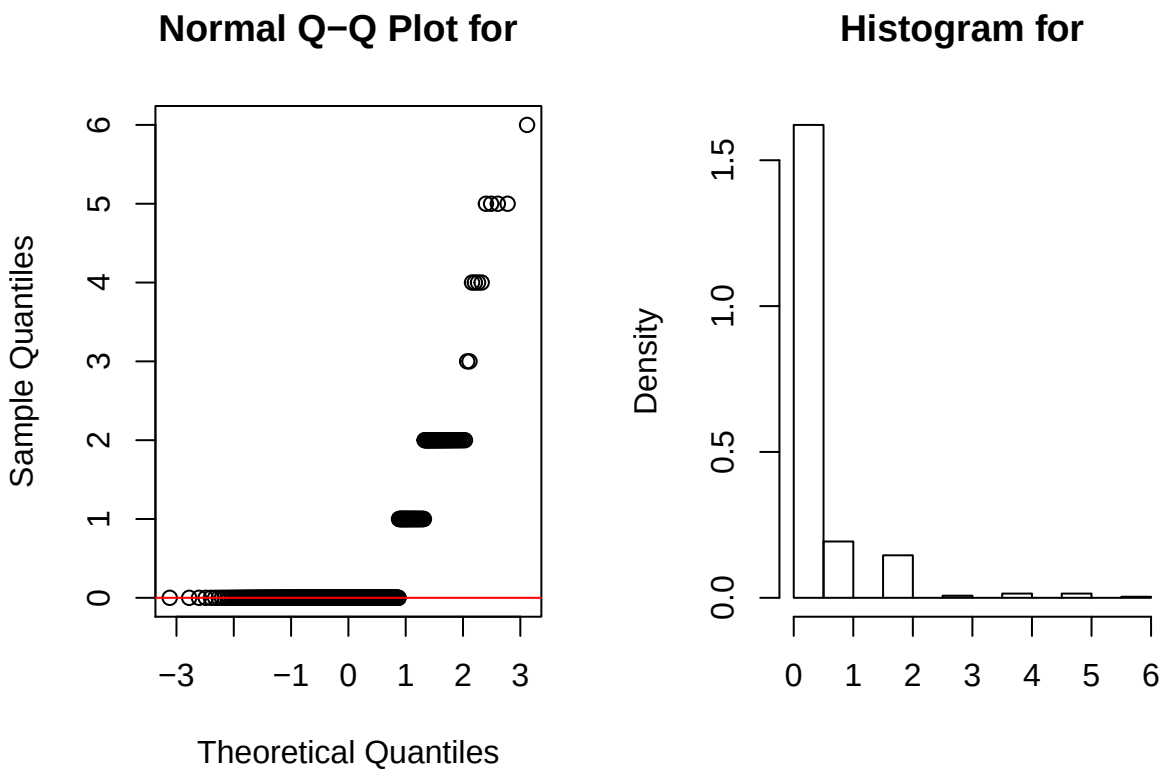
Dado los resultados anteriores, se observa que para las cuatro características consideradas sus correspondientes p-valores son inferiores al nivel de significación ($\alpha = 0.05$). Por tanto, rechazamos la H_0 y concluimos con un 95% de confianza que los datos no se distribuyen normalmente.

Normalidad de Supervivencia y Número de padres/hijos a bordo (Parch)

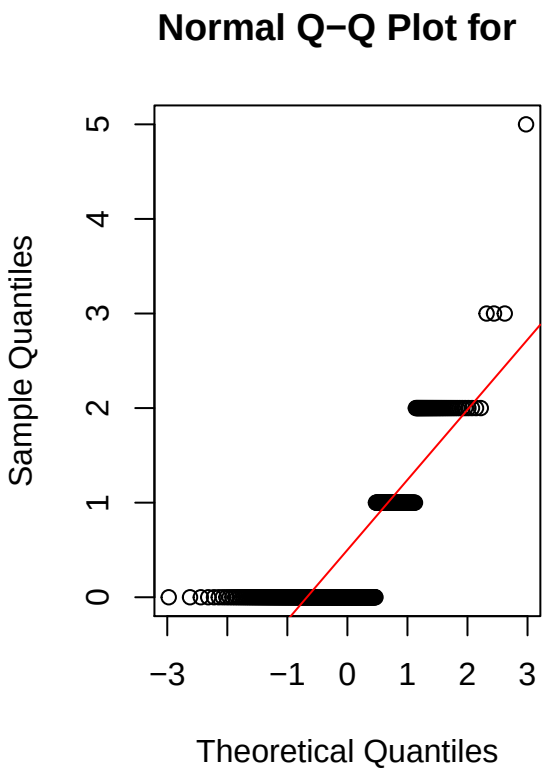
Ahora se aplicará este test para realizar el contraste de si existen diferencias en la característica Número de padres/hijos a bordo (Parch) en función de la supervivencia (Survived).

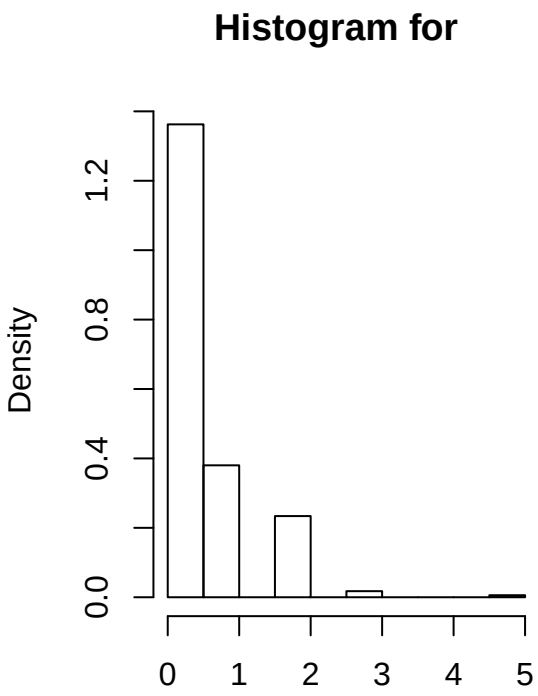
```
Parch_sur_0 <- clean_train$Parch[clean_train$Survived==0]
Parch_sur_1 <- clean_train$Parch[clean_train$Survived==1]
```

```
par(mfrow=c(1,2))
qqnorm(Parch_sur_0, main = paste("Normal Q-Q Plot for ", colnames(Parch_sur_0)[1]))
qqline(Parch_sur_0, col="red")
hist(Parch_sur_0,
  main=paste("Histogram for ", colnames(Parch_sur_0)[1]),
xlab=colnames(Parch_sur_0)[1], freq = FALSE)
```




```
par(mfrow=c(1,2))
qqnorm(Parch_sur_1, main = paste("Normal Q-Q Plot for ", colnames(Parch_sur_1)[1]))
qqline(Parch_sur_1, col="red")
hist(Parch_sur_1,
  main=paste("Histogram for ", colnames(Parch_sur_1)[1]),
xlab=colnames(Parch_sur_1)[1], freq = FALSE)
```






```
# Test Shapirp para Fare
shapiro.test(Parch_sur_0)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Parch_sur_0
## W = 0.45882, p-value < 2.2e-16
```

```
# Test Shapirp para Fare
shapiro.test(Parch_sur_1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Parch_sur_1
## W = 0.63887, p-value < 2.2e-16
```

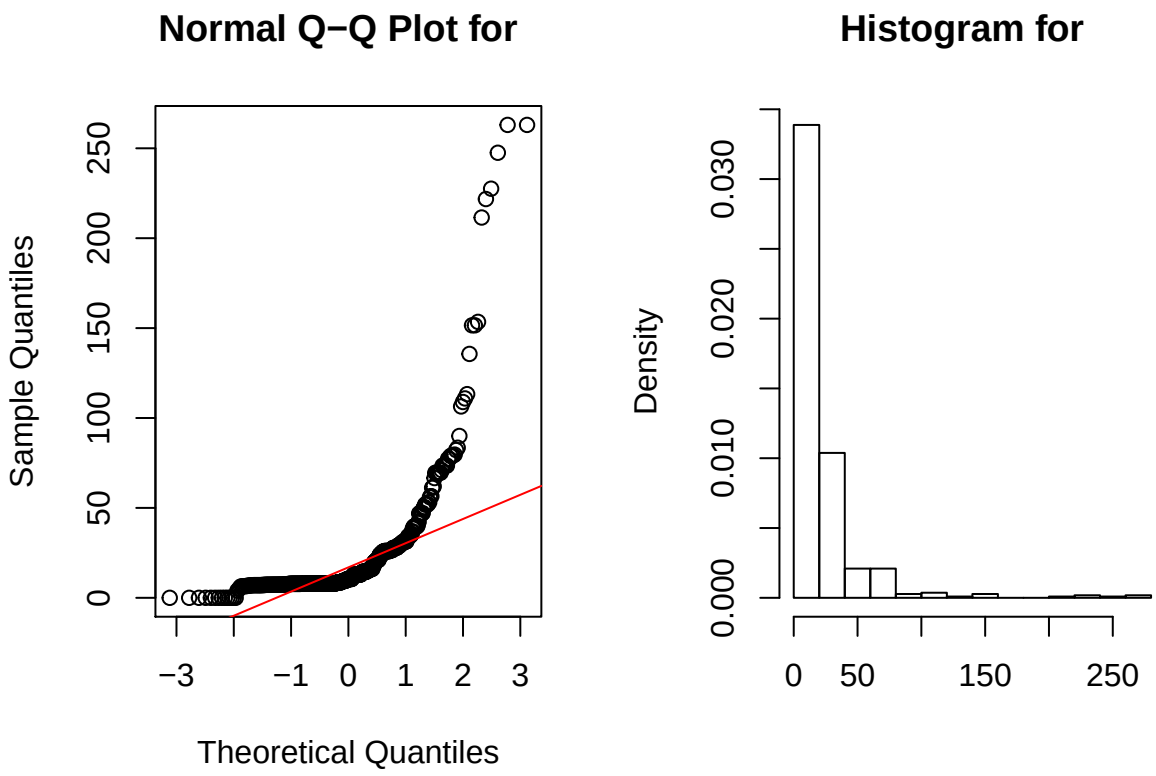
Dado los resultados anteriores, se observa que para las cuatro características consideradas sus correspondientes p-valores son inferiores al nivel de significación ($\alpha = 0.05$). Por tanto, rechazamos la H_0 y concluimos con un 95% de confianza que los datos no se distribuyen normalmente.

Normalidad de Supervivencia y Tarifa (Fare)

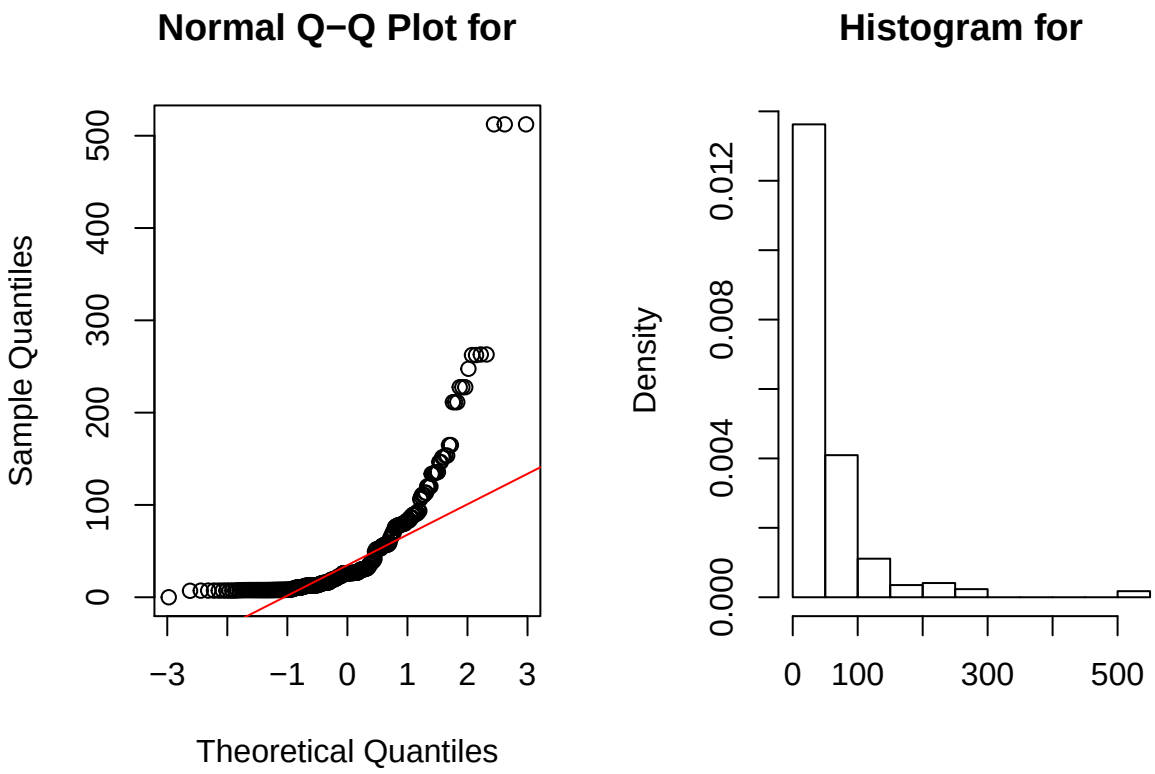
Ahora se aplicará este test para realizar el contraste de si existen diferencias en la característica Tarifa (Fare) en función de la supervivencia (Survived).

```
Fare_sur_0 <- clean_train$Fare[clean_train$Survived==0]
Fare_sur_1 <- clean_train$Fare[clean_train$Survived==1]
```

```
par(mfrow=c(1,2))
qqnorm(Fare_sur_0, main = paste("Normal Q-Q Plot for ", colnames(Fare_sur_0)[1]))
qqline(Fare_sur_0, col="red")
hist(Fare_sur_0,
  main=paste("Histogram for ", colnames(Fare_sur_0)[1]),
xlab=colnames(Fare_sur_0)[1], freq = FALSE)
```




```
par(mfrow=c(1,2))
qqnorm(Fare_sur_1, main = paste("Normal Q-Q Plot for ", colnames(Fare_sur_1)[1]))
qqline(Fare_sur_1, col="red")
hist(Fare_sur_1,
  main=paste("Histogram for ", colnames(Fare_sur_1)[1]),
xlab=colnames(Fare_sur_1)[1], freq = FALSE)
```




```
# Test Shapirp para Fare
shapiro.test(Fare_sur_0)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Fare_sur_0
## W = 0.51304, p-value < 2.2e-16
```

```
# Test Shapirp para Fare
shapiro.test(Fare_sur_1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Fare_sur_1
## W = 0.59673, p-value < 2.2e-16
```

Dado los resultados anteriores, se observa que para las cuatro características consideradas sus correspondientes p-valores son inferiores al nivel de significación ($\alpha = 0.05$). Por tanto, rechazamos la H_0 y concluimos con un 95% de confianza que los datos no se distribuyen normalmente.

4.2.2. Homogeneidad de la Varianza

Para estudiar la homogeneidad de varianzas se utiliza el test de Fligner-Killeen. Se trata de un test no paramétrico que compara las varianzas basándose en la mediana. Es una alternativa cuando no se cumple la condición de normalidad en las muestras. De tal forma que la hipótesis nula (H_0) y la alternativa (H_1) se pueden escribir de la siguiente forma:

Hipótesis nula (H_o): Todas las varianzas de las poblaciones son iguales.

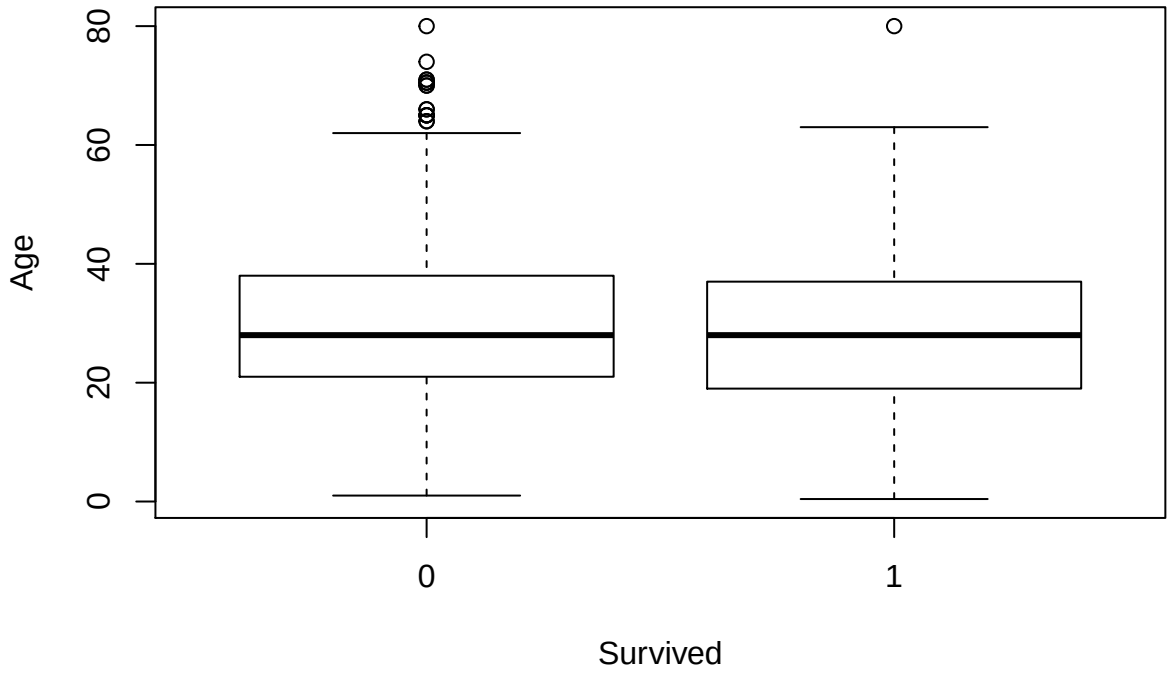
Hipótesis alternativa (H_1): Al menos dos de ellos difieren.

Zona de rechazo. Para todo valor de probabilidad mayor que un nivel de significación $\alpha = 0.05$, se acepta H_o y se rechaza H_1.

Para realizar el test Fligner-Killeen se utiliza la función **`fligner.test()`**.

A continuación, se muestra la aplicación del test Fligner-Killeen para la característica cuantitativas Edad (Age) en función de la Supervivencia (Survived):

```
# Test fligner para Age
boxplot(Age ~ Survived, data = clean_train)
```




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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 2.2627, df = 1, p-value = 0.1325
```

Puesto que obtenemos un p-valor superior al nivel de significación ($\alpha = 0.05$), aceptamos la hipótesis nula (H_0), es decir, de que las varianzas de ambas muestras son homogéneas.

A continuación, se muestra la aplicación del test Fligner-Killeen para característica Número de hermanos/cónyuges a bordo (SibSp) en función de la Supervivencia (Survived):

```
# Test fligner para SibSp
fligner.test(SibSp ~ Survived, data = clean_train)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  SibSp by Survived
## Fligner-Killeen:med chi-squared = 1.2514, df = 1, p-value = 0.2633
```

Puesto que obtenemos un p-valor superior al nivel de significación ($\alpha = 0.05$), aceptamos la hipótesis nula (H_0), es decir, de que las varianzas de ambas muestras son homogéneas.

A continuación, se muestra la aplicación del test Fligner-Killeen para la característica Número de padres/hijos a bordo (Parch) en función de la Supervivencia (Survived):

```
# Test fligner para Parch
fligner.test(Parch ~ Survived, data = clean_train)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Parch by Survived
## Fligner-Killeen:med chi-squared = 11.253, df = 1, p-value =
## 0.0007948
```

Puesto que obtenemos un p-valor superior al nivel de significación ($\alpha = 0.05$), aceptamos la hipótesis nula (H_0), es decir, de que las varianzas de ambas muestras son homogéneas.

A continuación, se muestra la aplicación del test Fligner-Killeen para la característica Tarifa (Fare) en función de la Supervivencia (Survived):

```
# Test fligner para Fare
fligner.test(Fare ~ Survived, data = clean_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
```

Puesto que obtenemos un p-valor inferior al nivel de significación ($\alpha = 0.05$), rechazamos la hipótesis nula (H_0), y podemos concluir que las varianzas son significativamente diferentes.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.

```
# Nivel de significancia
sig_level = 0.05
```

4.3.1. ¿Qué variables cuantitativas influyen más en el supervivencia?

En este apartado se aplicará un contraste de hipótesis sobre dos muestras para determinar si la supervivencia dependiendo de otra variable categórica. Para comparar la dependencia entre dos variables categóricas se utilizará la prueba de χ^2 (chi-cuadrado).

El contraste de hipótesis a realizar se expresa así:

Hipótesis nula (H_o). Los dos factores son independientes.

Hipótesis alternativa (H_1): Los dos factores son dependentes.

Zona de rechazo. Para todo valor de probabilidad mayor que un nivel de significación $\alpha = 0.05$, se acepta H_o y se rechaza H_1.

Una vez establecido las hipótesis para cada conjunto de variables categóricas consideradas se construirá su correspondiente tabla de contingencia y se aplicará el test chi-cuadrado, para ello se empleará la función `chisq.test()`.

A continuación, se calculan la prueba χ^2 para varios pares de variables categóricas.

Supervivencia vs Sexo (Sex)

```
tbl = table(clean_train$Survived, clean_train$Sex)
tbl
```

```
##
##       F   M
##   0  81 468
##   1 233 109
```

Aplicamos la función chisq.test a la tabla de contingencia tbl

```
## Test chi
chisq.test(tbl)
```

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

Como el valor de p-valor es menor que el nivel de significancia de 0.05, por tanto rechazamos la hipótesis nula (H_0) y aceptamos la hipótesis alternativa. Por tanto, concluimos que la supervivenvia depende del sexo del pasajero (Sex)

Supervivencia vs Clase (Pclass)

```
tbl = table(clean_train$Survived, clean_train$Pclass)
tbl
```

```
##
##       1   2   3
##   0  80  97 372
##   1 136  87 119
```

Aplicamos la función chisq.test a la tabla de contingencia tbl

```
## Test chi
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 102.89, df = 2, p-value < 2.2e-16
```

Como el valor de p-valor es menor que el nivel de significancia de 0.05, por tanto rechazamos la hipótesis nula (H_0) y aceptamos la hipótesis alternativa. Por tanto, concluimos que la supervivenvia depende la clase del pasajero (Pclass).

Supervivencia vs Clase (FsizeD)

```
tbl = table(clean_train$Survived, clean_train$FsizeD)
tbl
```

```
##
##     large singleton small
##   0    52       374   123
##   1    10       163   169
```

Aplicamos la función chisq.test a la tabla de contingencia tbl

```
## Test chi
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 74.537, df = 2, p-value < 2.2e-16
```

Como el valor de p-valor es menor que el nivel de significancia de 0.05, por tanto rechazamos la hipótesis nula (H_0) y aceptamos la hipótesis alternativa. Por tanto, concluimos que la supervivenvia depende del tamaño de la familia (FsizeD)

Supervivencia vs Titulo del Pasajero (Titulo)

```
tbl = table(clean_train$Survived, clean_train$Title)
tbl
```

```
##
##     Master Miss  Mr Mrs Rare Title
##   0     17   55 436  26         15
##   1     23  130  81 100          8
```

Aplicamos la función chisq.test a la tabla de contingencia tbl

```
## Test chi
chisq.test(tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 288.12, df = 4, p-value < 2.2e-16
```

Como el valor de p-valor es menor que el nivel de significancia de 0.05, por tanto rechazamos la hipótesis nula (H_0) y aceptamos la hipótesis alternativa. Por tanto, concluimos que la supervivenvia depende del titulo del pasajero (Title)

Supervivencia vs Cubierta del Camarote (Deck)

```
tbl = table(clean_train$Survived, clean_train$Deck)
tbl
```

```
##
##       A   B   C   D   E   F   G   T  U1  U2  U3
##   0   8  12  24   8   8   5   2   1  21  94 366
##   1   7  35  35  25  24   8   2   0  19  74 113
```

Aplicamos la función chisq.test a la tabla de contingencia tbl

```
## Test chi
chisq.test(tbl)
```

```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 126.68, df = 10, p-value < 2.2e-16
```

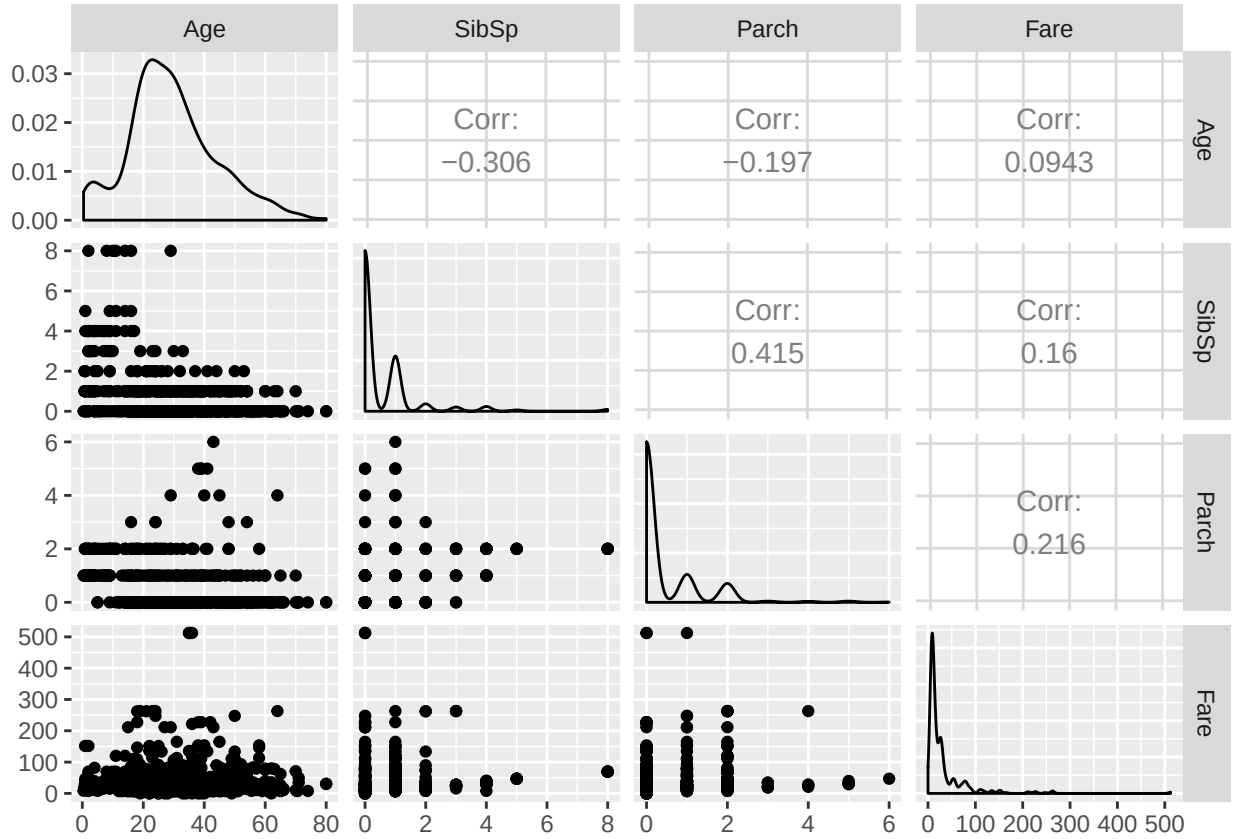
Como el valor de p-valor es menor que el nivel de significancia de 0.05, por tanto rechazamos la hipótesis nula (H_0) y aceptamos la hipótesis alternativa. Por tanto, concluimos que la supervivenvia depende de la cubierta (Deck)

4.3.2. Correlaciones

En este apartado procedemos a realizar un análisis de correlación entre las distintas variables numéricas del conjunto de datos.

Cuando dos características o más tienen correlación, eso significa que se están explicando unas a otras al tiempo con lo que proporcionan solo poca o ninguna información nueva.

```
# Calculamos las correlaciones.
corr_data <- select_if(clean_train, is.numeric)
corr_data <- select(corr_data, -PassengerId)

corr.res <- cor(corr_data)

# Mostramos las gráficas
ggpairs(corr_data)
```



La gráfica anterior muestra que existe una correlación positiva entre las variables **Parch** y **SibSp**. Esto tiene sentido debido a que ambas variables hacen referencia al tamaño de la familia que va a bordo.

4.3.2. Regresión Lineal logística.

Dado que la variable resultado (dependiente) solo pueda tomar dos valores (1=Vivo y 0=Muerto), la regresión logística será más adecuada que la regresión lineal.

La **regresión logística** es un tipo de análisis de regresión utilizado para predecir el resultado de una variable dicotómica dependiente, en función de una serie de variables independientes o predictoras. Dado que este modelo estima las probabilidades de ocurrencia, en lugar de utilizar un modelo aditivo que podría predecir valores fuera del rango [0,1], utiliza una escala transformada basada en una función logística

La estrategia por seguir será partir de un modelo donde la supervivencia dependa de la Edad (Age), la tarifa (Fare), el Número de hermanos/cónyuges a bordo (SibSp), el número de padres/hijos a bordo (Parch), la embarcación (Embarked), el sexo (Sex) y la clase (Pclass). Partiendo de esta modelo ser irá añadiendo y quitando variables con el propósito de mejorar el modelo.

En primer lugar, establecemos categorías de referencia para las variables cualitativas: "F" para la variable Sex, "S" para la variable Embarked, "1" para la variable Pclass, "small" para la variable Fsize, "Miss" para la variable Title, y "A" para la variable Title; para ello utilizamos la función `relevel()`.

```
# Nivel de significancia
sig_level = 0.05

# Establecemos categoria de referencia conjunto de datos.
clean_train$SexR <- relevel(clean_train$Sex, ref="F")
clean_train$EmbarkedR <- relevel(clean_train$Embarked, ref="S")
clean_train$PclassR <- relevel(clean_train$Pclass, ref="1")
clean_train$FsizeD <- relevel(clean_train$Fsize, ref="small")
clean_train$TitleR <- relevel(clean_train$Title, ref="Miss")
clean_train$DeckR <- relevel(clean_train$Deck, ref="A")

# Establecemos categoria de referencia conjunto de pruebas
clean_test$SexR <- relevel(clean_test$Sex, ref="F")
clean_test$EmbarkedR <- relevel(clean_test$Embarked, ref="S")
clean_test$PclassR <- relevel(clean_test$Pclass, ref="1")
clean_test$FsizeD <- relevel(clean_test$Fsize, ref="small")
clean_test$TitleR <- relevel(clean_test$Title, ref="Miss")
clean_test$DeckR <- relevel(clean_test$Deck, ref="A")
```

Calculamos la supervivencia en función de las carácteristicas:

- Modelo 1. Survived = Age + SibSp + Parch + Fare + EmbarkedR + SexR + PclassR.
- Modelo 2. Survived = Age + SibSp + Parch + Fare + EmbarkedR + SexR + PclassR + FsizeD.
- Modelo 3. Survived = Age + SibSp + Parch + Fare + EmbarkedR + SexR + PclassR + TitleR
- Modelo 4. Survived = Age + SibSp + Parch + Fare + EmbarkedR + SexR + PclassR + DeckR
- Modelo 5. Survived = Age + SibSp + Parch + Fare + EmbarkedR + SexR + PclassR + FsizeD + TitleR + DeckR
- Modelo 6. Survived = Age + Fare + SexR + PclassR + FsizeD + TitleR
- Modelo 7. Survived = Age + Fare + SexR + PclassR + FsizeD + DeckR
- Modelo 8. Survived = Age + SexR + PclassR + FsizeD

```
# Calculamos modelo
glm1.fit <- glm(factor(Survived) ~ Age + SibSp + Parch + Fare +
                                   EmbarkedR + SexR + PclassR,
                data = clean_train,
                family = "binomial")
```

```r
# Obtenemos resumen
glm1.summary <- summary(glm1.fit)
#glm1.summary
```

```r
# Calculamos modelo
glm2.fit <- glm(factor(Survived) ~ Age + SibSp + Parch + Fare +
                                   EmbarkedR + SexR + PclassR +
                                   FsizeD,
                data = clean_train,
                family = "binomial")
# Obtenemos resumen
glm2.summary <- summary(glm2.fit)
#glm2.summary
```

```r
# Calculamos modelo
glm3.fit <- glm(factor(Survived) ~ Age + SibSp + Parch + Fare +
                                   EmbarkedR + SexR + PclassR +
                                   TitleR,
                data = clean_train,
                family = "binomial")
# Obtenemos resumen
glm3.summary <- summary(glm3.fit)
#glm3.summary
```

```r
# Calculamos modelo
glm4.fit <- glm(factor(Survived) ~ Age + SibSp + Parch + Fare +
                                   EmbarkedR + SexR + PclassR +
                                   DeckR,
                data = clean_train,
                family = "binomial")
# Obtenemos resumen
glm4.summary <- summary(glm4.fit)
#glm4.summary
```

```r
# Calculamos modelo
glm5.fit <- glm(factor(Survived) ~ Age + SibSp + Parch + Fare +
                                   EmbarkedR + SexR + PclassR +
                                   FsizeD + TitleR + DeckR,
                data = clean_train,
                family = "binomial")

# Obtenemos resumen
glm5.summary <- summary(glm5.fit)
#glm5.summary
```

```r
# Calculamos modelo
glm6.fit <- glm(factor(Survived) ~ Age + Fare +
                                   SexR + PclassR +
                                   FsizeD + TitleR,
                data = clean_train,
                family = "binomial")

# Obtenemos resumen
glm6.summary <- summary(glm6.fit)
#glm6.summary
```

```r
# Calculamos modelo
glm7.fit <- glm(factor(Survived) ~ Age + Fare +
                                   SexR + PclassR +
                                   FsizeD + DeckR,
                data = clean_train,
                family = "binomial")

# Obtenemos resumen
glm7.summary <- summary(glm7.fit)
#glm7.summary
```

```r
# Calculamos modelo
glm8.fit <- glm(factor(Survived) ~ Age +
                                   SexR + PclassR +
                                   FsizeD,
                data = clean_train,
                family = "binomial")

# Obtenemos resumen
glm8.summary <- summary(glm8.fit)
#glm8.summary
```

Para los anteriores modelos de regresión logística obtenidos, la bondad del modelo se evaluará mediante la medida AIC (criterio de información de Akaike, por sus siglas en inglés Akaike Information Criterion). Dado que esta medida tiene en cuenta tanto la bondad del ajuste como la complejidad del modelo, cuando se comparen varios modelos candidatos, se seleccionará aquel que resulte en el menor AIC. Para obtener los AIC's de los modelos se utiliza la función `AIC()`.

```r
aui_data <- AIC(glm1.fit, glm2.fit, glm3.fit, glm4.fit, glm5.fit, glm6.fit, glm7.fit, glm8.fit)
kable(aui_data) %>%
  kable_styling(bootstrap_options = "striped", full_width = F)
```

	df	AIC
glm1.fit	10	809.0809
glm2.fit	12	792.8541
glm3.fit	14	755.6265
glm4.fit	20	811.7659
glm5.fit	26	753.9042
glm6.fit	12	741.5767
glm7.fit	18	794.0856
glm8.fit	7	789.7273

Dado los resultados anteriores se llega a la conclusión que se obtiene el mejor resultado con el modelo regresor 4 con un valor de 811.766.

```r
glm4.summary
```

```
##
## Call:
## glm(formula = factor(Survived) ~ Age + SibSp + Parch + Fare +
##     EmbarkedR + SexR + PclassR + DeckR, family = "binomial",
##     data = clean_train)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.4558  -0.6022  -0.3983   0.6117   2.4677
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.502094   0.685921   5.106 3.30e-07 ***
## Age          -0.033025   0.007567  -4.364 1.28e-05 ***
## SibSp        -0.340306   0.109512  -3.107  0.00189 **
## Parch        -0.097982   0.119297  -0.821  0.41146
## Fare          0.002707   0.002681   1.010  0.31252
## EmbarkedRC    0.516209   0.244584   2.111  0.03481 *
## EmbarkedRQ    0.316817   0.340829   0.930  0.35261
## SexRM        -2.699291   0.205361 -13.144  < 2e-16 ***
## PclassR2      0.143647   1.109066   0.130  0.89695
## PclassR3     -1.023828   1.174689  -0.872  0.38344
## DeckRB        0.029073   0.704977   0.041  0.96711
## DeckRC       -0.344894   0.657580  -0.524  0.59994
## DeckRD        0.489003   0.738516   0.662  0.50788
## DeckRE        0.919261   0.745893   1.232  0.21779
## DeckRF       -0.018349   1.327755  -0.014  0.98897
## DeckRG       -1.735648   1.661714  -1.044  0.29626
## DeckRT      -12.978863 535.411462  -0.024  0.98066
## DeckRU1      -0.686211   0.680247  -1.009  0.31309
## DeckRU2      -1.209092   1.248041  -0.969  0.33265
## DeckRU3      -1.203756   1.298147  -0.927  0.35378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  771.77  on 871  degrees of freedom
## AIC: 811.77
##
## Number of Fisher Scoring iterations: 12
```

```r
glm4.coef <- coef(glm4.fit)
glm4.coef_exp <- exp(coef(glm1.fit))
data <- data.frame(Coeficiente = glm4.coef, Exp = glm4.coef_exp)
kable(data) %>%
  kable_styling(bootstrap_options = "striped", full_width = F)
```

	Coeficiente	Exp
(Intercept)	3.5020940	32.9949208
Age	-0.0330247	0.9673129
SibSp	-0.3403063	0.7080273
Parch	-0.0979818	0.9191893
Fare	0.0027072	1.0020653
EmbarkedRC	0.5162094	1.5765585
EmbarkedRQ	0.3168167	1.3733814
SexRM	-2.6992913	0.0686356
PclassR2	0.1436467	0.3912136
PclassR3	-1.0238279	0.1121338
DeckRB	0.0290726	32.9949208
DeckRC	-0.3448941	0.9673129
DeckRD	0.4890031	0.7080273
DeckRE	0.9192611	0.9191893
DeckRF	-0.0183485	1.0020653
DeckRG	-1.7356476	1.5765585
DeckRT	-12.9788628	1.3733814
DeckRU1	-0.6862106	0.0686356
DeckRU2	-1.2090918	0.3912136
DeckRU3	-1.2037563	0.1121338

Del modelo anterior podemos concluir que:

- Para las variables Age, SibSp, EmbarkedRC, y SexRM, sus correspondientes p-valores son menores que 0.05, es decir, son significativas para el modelo.
- Para las variables Parch, Fare, EmbarkedRQ, PclassR2, PclassR3, DeckRB, DeckRC, DeckRD, DeckRE, DeckRF, DeckRG, DeckRT, DeckRU1, DeckRU2 y DeckRU3 su correspondientes p-valores son mayores que 0.05. Por tanto, no son estadísticamente significativas para el resultado y se pueden eliminar del modelo.

Predicción.

Ahora, empleando este modelo podemos proceder a realizar predicciones de la supervivencia con el conjunto de prueba.

```
# Calculamos la probabilidad del conjunto de test
glm4.test_prob <- predict(glm4.fit, newdata = clean_test, type = "response")

# Calculamos un prediccion
test_threshold <- 0.75
glm4.test_pred <- ifelse(glm4.test_prob > test_threshold, 1, 0)

# Generamos la matriz de confusion
glm4.confusionMatrix <- confusionMatrix(data=factor(glm4.test_pred),
                                        reference=factor(clean_test$Survived))
glm4.confusionMatrix
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0 264  79
##          1   2  73
```

```
##
##                Accuracy : 0.8062
##                  95% CI : (0.765, 0.843)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : 2.765e-14
##
##                   Kappa : 0.5303
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9925
##             Specificity : 0.4803
##          Pos Pred Value : 0.7697
##          Neg Pred Value : 0.9733
##              Prevalence : 0.6364
##          Detection Rate : 0.6316
##    Detection Prevalence : 0.8206
##       Balanced Accuracy : 0.7364
##
##        'Positive' Class : 0
##
```

Interpretación.

De los resultados anteriores podemos concluir que:

- El número de **falso positivos** es 2. La tasa falsos positivos es del 51.97 %.
- El número de **falso negativos** es 79. La tasa falsos negativos es del 0.75 %.
- La **exactitud** es del 80.6220096 %. La tasa de error es 19.3779904 %
- La **sensibilidad** es del 99.25 %.
- La **especificidad** es del 48.03 %.

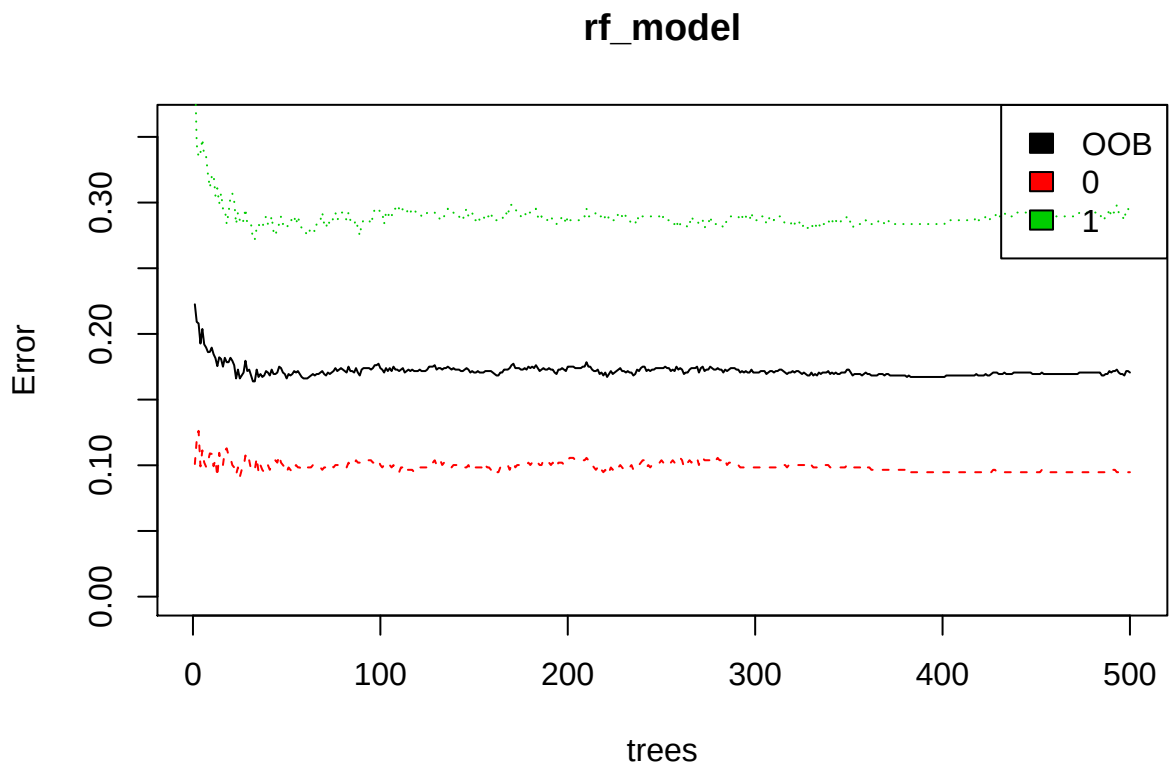
4.3.1. Random Forests

¿Qué variables son las más importantes para nuestro modelo de clasificación?

Un método habitual utilizado para responder esta pregunta es **Random Forest** (RF). El **RF** es un método de clasificación basado en la realización de múltiples árboles de decisión sobre muestras de un conjunto de datos Además, Random Forest permite obtener medidas acerca de la importancia que los diferentes predictores han tenido en el modelo, lo que permite en parte interpretar este. La importancia de los predictores se evalúa como el número de veces que han sido utilizados por los diversos árboles y su capacidad para reducir el índice de Gini en ellos

```
# Set a random seed
set.seed(754)

# Build the model (note: not all possible variables are used)
rf_model <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch +
                                            Fare + Embarked + Title +
                                            FsizeD,
                                            data = clean_train)

# Show model error
```

```r
plot(rf_model, ylim=c(0,0.36))
legend('topright', colnames(rf_model$err.rate), col=1:3, fill=1:3)
```



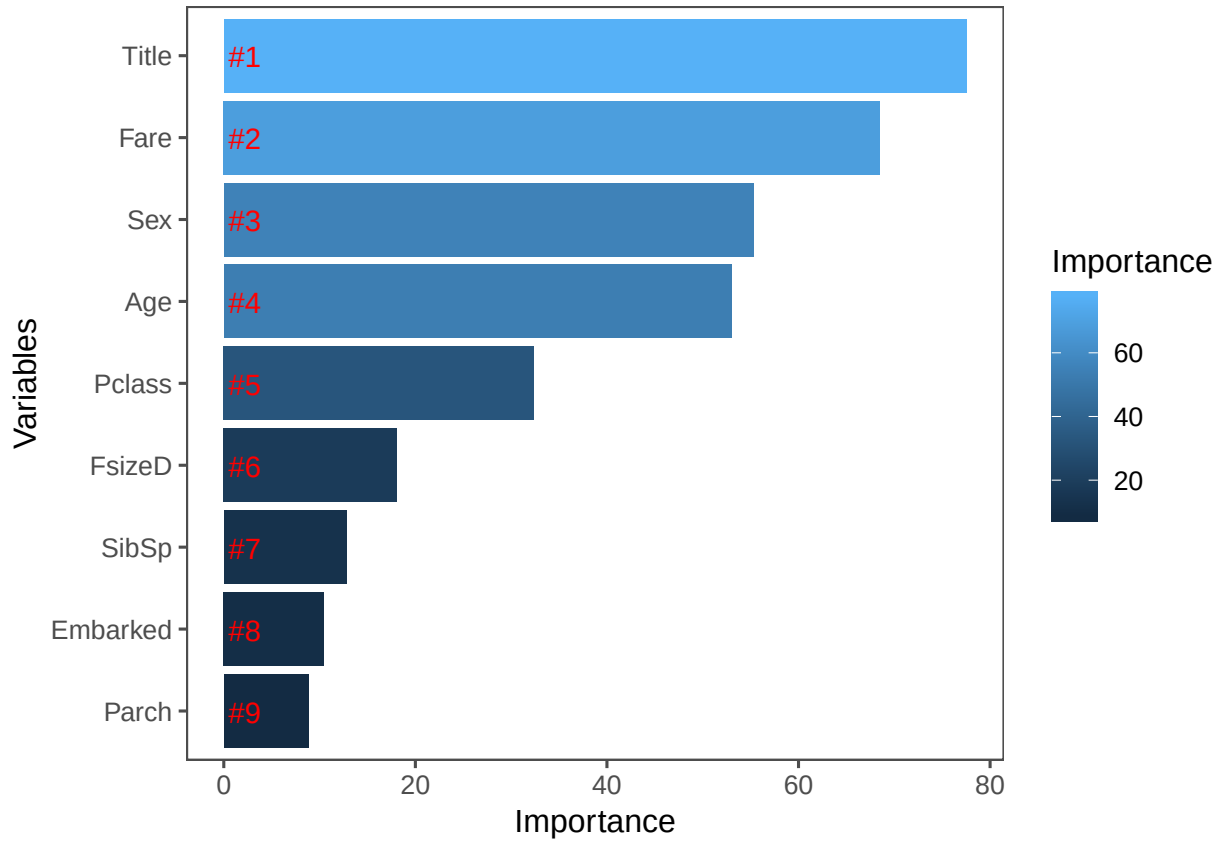
La línea negra muestra la tasa de error general que cae por debajo del 20%. Las líneas rojas y verdes muestran la tasa de error de "muerto" y "sobrevivió" respectivamente. Con alrededor del 10%, nuestro modelo parece ser bueno para predecir mejor la muerte que la supervivencia.

Importancia de las variables.

Ahora, veamos la importancia de la variable relativa al explorar la disminución media en Gini calculada en todos los árboles.

```r
# Get importance
importance    <- importance(rf_model)
varImportance <- data.frame(Variables = row.names(importance),
                            Importance = round(importance[ ,'MeanDecreaseGini'],2))

# Create a rank variable based on importance
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))

# Use ggplot2 to visualize the relative importance of variables
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
    y = Importance, fill = Importance)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = Variables, y = 0.5, label = Rank),
    hjust=0, vjust=0.55, size = 4, colour = 'red') +
```

```
  labs(x = 'Variables') +
  coord_flip() +
  theme_few()
```



En la gráfica anterior se observa que las variables Title y Fare se consideran las más importantes. Esto contradice al método de regresión logística que las consideraba no significativa. Por otra parte, la variable SibSp está clasificado en séptimo lugar; mientras en la regresión logística era estadísticamente significativa. Sin embargo, la variable FsizeD clasifica mejor que las variables SibSp y Parch. Esto tiene sentido ya que FsizeD es la discretización de la combinación de estas dos variables.

Predicción.

```
# Predict using the test set
forest.test_pred <- predict(rf_model, clean_test)

# Generamos la matriz de confusion
forest.confusionMatrix <- confusionMatrix(data=factor(forest.test_pred),
                                          reference=factor(clean_test$Survived))
forest.confusionMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 242  25
##          1  24 127
##
```

```
##                Accuracy : 0.8828
##                  95% CI : (0.848, 0.912)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7464
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9098
##             Specificity : 0.8355
##          Pos Pred Value : 0.9064
##          Neg Pred Value : 0.8411
##              Prevalence : 0.6364
##          Detection Rate : 0.5789
##    Detection Prevalence : 0.6388
##       Balanced Accuracy : 0.8727
##
##        'Positive' Class : 0
##
```

Interpretación.

De los resultados anteriores podemos concluir que:

- El número de **falso positivos** es 24. La tasa falsos positivos es del 16.45 %.
- El número de **falso negativos** es 25. La tasa falsos negativos es del 9.02 %.
- La **exactitud** es del 88.277512 %. La tasa de error es 11.722488 %
- La **sensibilidad** es del 90.98 %.
- La **especificidad** es del 83.55 %.

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

Durante el desarrollo de este trabajo en los apartados anteriores de mostraron los resultados obtenidos mediante diagramas de barras, boxplot y tablas.

6. Resolución del problema.

En este trabajo se trató de la problemática de determinar qué variables influyeron más sobre la supervivencia de los pasajeros a bordo del Titanic. Para llevar a cabo esta tarea se realizó se utilizó el *conjunto de datos de entrenamiento* y *conjunto de datos de prueba*.

Sobre este conjunto de datos se realizó una fase de preprocesamiento que incluye varias tareas de limpieza de datos, (tales como, conversiones, eliminación los valores perdidos o nulos), discretización de valores numéricos, etc. En la imputación de valores perdidos se pueden destacar el trabajo realizado en las variables Edad (Age) y Cubierta (Deck).

Para la imputación de los valores perdidos de la característica edad (Age) se empleó el algoritmo **MICE** (Multivariate Imputation by Chained Equations) de R. MICE se ha convertido en un método de referencia para tratar los datos perdidos.

La característica Deck se creó a partir del primer carácter de la variable Cabina (Cabin). Este carácter hace referencia al nivel de cubierta en el que estaba ubicada cabina. Las cabinas de una clase más alta estaban

más cerca de la cubierta principal y de los botes salvavidas. Sin embargo, la característica Deck es la que más valores desconocidos presenta en el conjunto de datos. Debido a su gran número no se pueden eliminar las filas que tengan valores perdidos para este campo. Para imputar sus valores se utilizó los valores $U1$, $U2$ y $U3$, estos valores se asignan en función de la clase del pasajero.

Se realizaron pruebas estadísticas para comprobar las dependencias entre Supervivencia y otras variables categóricas del conjunto de datos. Se identifico que existen evidencias estadísticas de dependencias entre la Supervivencia y las variables categóricas: Sexo (Sex), Clase (Pclass), Tamaño de la familia (FsizeD), Titulo (Title) y Cubierta (Deck).

En este trabajo se utilizaron los clasificadores **Random Forest** (RF) y **Regresión Logística** (LR) con el propósito de construir un modelo que permita predecir la supervivencia a partir de un conjunto de pruebas.

El clasificador **Random Forest** también nos permite medir la importancia de las variables predictoras en el modelo de clasificación. El clasificador **RF** encontró que las variables *SibSp*, *Embarked* y *Parch* no parecen jugar un rol significante en la determinación de la supervivencia. Se podrían eliminar estas características para observar si existe una mejora en el modelo.

El clasificador de **Regresión Logística** también nos permite cuantificar la importancia de la relación existente entre cada una de las covariables y la variable dependiente. El clasificador **RF** encontró que las variables *Age*, *SibSp*, *EmbarkedRC*, y *SexRM* son significativas para el modelo.

En ambos modelos **RF** y **LR** no indican que las variables Age y Sex influyen en la supervivencia. Sin embargo, el modelo **RF** también considera importantes las variables Fare, Title y Pclass. Mientras que el modelo **LR** considera que las variables SibSp y EmbarkedRC son significativas. Al analizar estas discrepancias podríamos posicionarnos en el lado del modelo **RF** debido a que da mayor importancia a variables Fare y Pclass que son socioeconómicas y estas si pueden influir en la supervivencia.

Por otra parte, si comparamos los resultados obtenidos de la predicción entre el *LR* y el *RF* se observa que tienen una **exactitud** del 80.6220096 % y 88.277512 % respectivamente. Por tanto, el clasificador *RF* es superior al clasificador de *RL*.

Como tarea adicional se puede plantear discretizar la variable Edad (Age). Si tenemos en consideración que las mujeres y niños tienen prioridad en el momento de abordar los botes salvavidas. El discretizar esta variable (por ejemplo: niño, joven, adulto, y anciano) puede convertir esta nueva una variable que puede influir en la supervivencia.

Otro cambio podría incluir mejorar la creación de la variable Título. Por ejemplo, los valores 'Lady', 'Sir', y 'Jonkheer' se pueden mapear a un valor de 'Royalty' en lugar de 'Rare Title'. Este cambio resalta la condición socioeconómica del pasajero. Este factor puede influir en la supervivencia.

Contribuyentes

Contribuciones	Firma
Investigaciones previas	Edison Muzo
Redacción de las respuestas	Edison Muzo
Desarrollo código	Edison Muzo

References

Baayen, R Harald. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*. Cambridge University Press.

Buuren, S van, and Karin Groothuis-Oudshoorn. 2010. "Mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, 1–68.

Calvo, Mireia, Laia Subirats, and Diego Pérez. n.d. “Introducción a La Limpieza Y análisis de Los Datos.” *UOC*, 33.

datascienceplus. 2019. “Imputing Missing Data with R; Mice Package.” https://datascienceplus.com/imputing-missing-data-with-r-mice-package/. \url{https://datascienceplus.com/imputing-missing-data-with-r-mice-package/}.

Han, Jiawei, Micheline Kamber, and Data Mining. 2001. “Concepts and Techniques.” *Morgan Kaufmann* 340: 94104–93205.

Hothorn, Torsten, and Brian S Everitt. 2014. *A Handbook of Statistical Analyses Using R*. CRC press.

kaggle. 2019a. “Titanic: Machine Learning from Disaster: Overview.” https://www.kaggle.com/c/titanic/overview/tutorials. \url{https://www.kaggle.com/c/titanic/overview/tutorials}.

———. 2019b. “Titanic: Machine Learning from Disaster: Tutorial.” https://www.kaggle.com/c/titanic/overview. \url{https://www.kaggle.com/c/titanic/overview/tutorials}.

Osborne, Jason W. 2010. “Data Cleaning Basics: Best Practices in Dealing with Extreme Scores.” *Newborn and Infant Nursing Reviews* 10 (1): 37–43.

Risdal, Megan. 2019. “Exploring Survival on the Titanic.” https://www.kaggle.com/mrisdal/exploring-survival-on-the-titanic. \url{https://www.kaggle.com/mrisdal/exploring-survival-on-the-titanic}.

Squire, Megan. 2015. *Clean Data*. Packt Publishing Ltd.

Teetor, Paul. 2011. *R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics*. " O’Reilly Media, Inc.".

Williams, Steven. 2019. “Titanic Survival (Logistic Regression).” https://www.kaggle.com/drwilliamssteven/titanic-survival-logistic-regression. \url{https://www.kaggle.com/drwilliamssteven/titanic-survival-logistic-regression}.