

Práctica 2: Limpieza y análisis de datos

Paula de Jaime de Toro

2019-12-21

Table of Contents

1. Descripción del dataset.....	1
2. Integración y selección de los datos de interés a analizar.....	2
3. Limpieza de los datos.....	5
4. Análisis de los datos.....	14
5. Representación de los resultados a partir de tablas y gráficas.....	20
6. Conclusiones.....	25
7. Recursos.....	26

1. Descripción del dataset

El conjunto de datos seleccionado es “Titanic: Machine Learning from Disaster” y se encuentra en la plataforma “kaggle” (<https://www.kaggle.com/c/titanic>).

El 15 de abril de 1912 el Titanic se hundió tras chocar contra un iceberg. Desafortunadamente, no hubo suficientes lanchas salvavidas para todos los pasajeros, resultando esto en la muerte de 1502 pasajeros de un total de 2224.

Aunque la suerte influyó seguramente en la supervivencia de estos pasajeros, hay indicios de que ciertos grupos tenían más probabilidad de sobrevivir que otros.

El objetivo de este estudio será poder analizar las características que hacen que algunos pasajeros tengan más posibilidades de sobrevivir.

El dataset se compone de dos ficheros: un fichero de train (train.csv) y otro de de test (test.csv). Solo se utilizará el fichero “train.csv”.

Los atributos del dataset son:

- PassengerId: Identificador único de un pasajero.
- Survived: Indica si el pasajero sobrevivió (1) o no (0).
- Pclass: Indica la clase socio-económica del ticket que compró el pasajero (1=clase alta, 2=clase media y 3=clase baja).
- Name: Nombre del pasajero.
- Sex: Género del pasajero.
- Age: Edad del pasajero.

- SibSp: Número de hermanos, hermanastros, mujeres y cónyuges.
- Parch: Número de padres, hijos, hijastros. Algunos niños viajaron con una niñera sin padres, por lo que el valor en estos casos es 0.
- Ticket: El código del ticket.
- Fare: El precio del ticket.
- Cabin: El número de la cabina asignada.
- Embarked: El puerto donde subió abordo del Titanic (C=Cherbourg, Q=Queenstown y S=Southampton).

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

Lo primero que se hace es cargar los datos del fichero csv.

```
file_location <- "data/train.csv"
data <- read.csv(file_location)
```

El dataset importado cuenta con 891 registros y 12 atributos.

```
nrow(data)
## [1] 891

ncol(data)
## [1] 12
```

Un resumen de los datos:

```
## 'data.frame': 891 obs. of 12 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : Factor w/ 891 levels "Abbing, Mr. Anthony",...: 109 191
358 277 16 559 520 629 417 581 ...
## $ Sex : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1
1 ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : Factor w/ 681 levels "110152","110413",...: 524 597 670
50 473 276 86 396 345 133 ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : Factor w/ 148 levels "", "A10", "A14",...: 1 83 1 57 1 1
131 1 1 1 ...
## $ Embarked : Factor w/ 4 levels "", "C", "Q", "S": 4 2 4 4 4 3 4 4 4 2
...
```

2.1. Eliminar atributos

Antes de comprobar si el formato de los atributos es correcto, se van a borrar los atributos que no se vayan a necesitar en el análisis.

Se borra el identificador del pasajero (PassengerId) y el código del ticket (Ticket), ya que son atributos que no parecen importantes para saber si una persona sobrevivió o no.

```
data <- subset(data, select = -c(PassengerId, Ticket))

str(data)

## 'data.frame':    891 obs. of  10 variables:
## $ 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    : Factor w/ 891 levels "Abbing, Mr. Anthony",...: 109 191 35
## 8 277 16 559 520 629 417 581 ...
## $ Sex      : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 .
## ..
## $ Age      : num  22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp    : int  1 1 0 1 0 0 0 3 0 1 ...
## $ Parch    : int  0 0 0 0 0 0 0 1 2 0 ...
## $ Fare     : num  7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin    : Factor w/ 148 levels "", "A10", "A14",...: 1 83 1 57 1 1 131
## 1 1 1 ...
## $ Embarked: Factor w/ 4 levels "", "C", "Q", "S": 4 2 4 4 4 3 4 4 4 2 ..
## .
```

2.2. Formato variables

Variables categóricas

Las variables “Survived”, “Pclass”, “Sex” y “Embarked” serán variables categóricas al tomar valores dentro de un rango de categorías.

```
data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
data$Sex <- as.factor(data$Sex)
data$Embarked <- as.factor(data$Embarked)
```

Números

Las variables “Age”, “Fare”, “SibSp” y “Parch” serán números y se tratarán como tal.

```
data$Age <- as.integer(data$Age)
```

Carácteres

El atributo “Name” y “Cabin” serán de tipo string.

```
data$Name <- as.character(data$Name)
data$Cabin <- as.character(data$Cabin)
```

2.3. Añadir campos

En un primer momento se puede pensar que la gente que no tenía familia tenía más probabilidad de salvarse. Por este motivo, se incluye un campo “hasFamily” (0 o 1) que indica si un pasajero tiene familia. Un pasajero tendrá familia cuando alguna de las variables “SibSp” y “Parch” sea mayor de cero.

```
data$hasFamily[data$SibSp > 0 | data$Parch > 0] <- 1
data$hasFamily[is.na(data$hasFamily)] <- 0
data$hasFamily <- as.factor(data$hasFamily)
```

La información de si un pasajero tiene familia se guarda en el campo “hasFamily”, por lo que mantener los atributos “SibSp” y “Parch” ya no será necesario.

```
data <- subset(data, select = -c(SibSp, Parch))
```

Los nombres de los pasajeros tendrán un prefijo como “Mrs.” que indicarán el título que tienen. Se crea una columna con dichos valores. Serán valores categóricos.

```
namesWithTitle <- sapply(strsplit(as.character(data$Name), ","), "[[", 2)
data$Title <- sapply(strsplit(namesWithTitle), "\\.", "[[", 1)
data$Title <- as.factor(data$Title)
```

La columna del nombre del pasajero ya no será necesaria por lo que se borra.

```
data <- subset(data, select = -c(Name))
```

La estructura de los datos después de los cambios anteriores será:

```
str(data)

## 'data.frame': 891 obs. of 9 variables:
## $ 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 "female","male": 2 1 1 1 2 2 2 2 1 1 ...
## $ Age : int 22 38 26 35 35 NA 54 2 27 14 ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr "" "C85" "" "C123" ...
## $ Embarked : Factor w/ 4 levels "", "C", "Q", "S": 4 2 4 4 4 3 4 4 4 2 .
## $ hasFamily: Factor w/ 2 levels "0","1": 2 2 1 2 1 1 1 2 2 2 ...
## $ Title : Factor w/ 17 levels "Capt","Col","Don",...: 12 13 9 13 12 12 12 8 13 13 ...
```

3. Limpieza de los datos

Antes de tratar los elementos vacíos se comprueba que los datos sean correctos.

3.0. Preliminares

El campo “Survived” tiene dos valores: 0 (no sobrevive) y 1 (sobrevive).

```
levels(data$Survived)
## [1] "0" "1"
```

El campo “Pclass” tiene tres valores: 1=clase alta, 2=clase media y 3=clase baja.

```
levels(data$Pclass)
## [1] "1" "2" "3"
```

El sexo de los pasajeros es femenino o masculino.

```
levels(data$Sex)
## [1] "female" "male"
```

No hay nadie que tenga menos de 0 años o más de 100. Las personas más mayores tendrán 80 años, y las más pequeñas serán bebés con 0 años.

```
max(data$Age, na.rm = TRUE)
## [1] 80
min(data$Age, na.rm = TRUE)
## [1] 0
```

En el precio “Fare” el único valor extraño que se ha encontrado es el 0. Suponiendo que un billete no puede costar 0 euros y que estos no se ganaron, se interpretarán estos como valores perdidos.

```
data$Fare[data$Fare == 0] <- NA
```

En el atributo “Embarked” se observa que uno de los valores es “”, que significa valor vacío o NA. Los valores se sustituyen por NA y se actualizan los factores.

```
data$Embarked[data$Embarked == ""] <- NA
data$Embarked <- droplevels(data$Embarked)
```

Los valores vacíos de “Cabin” se sustituyen por un NA.

```
data$Cabin[data$Cabin == ""] <- NA
```

Se comprueba que las personas con título de mujer sean mujeres y que las personas con título de hombre sean hombres. Antes se van a procesar los títulos para comprobar que son correctos.

Los títulos actuales son:

```
table(data$Title)
```

```
##
##      Capt      Col      Don      Dr      Jonkheer
##      1        2        1        7        1
##      Lady      Major      Master      Miss      Mlle
##      1        2        40       182        2
##      Mme       Mr       Mrs       Ms       Rev
##      1       517      125        1        6
##      Sir the Countess
##      1        1
```

- Madame (Mme) es lo mismo que Mrs, y se referirá a mujeres casadas.
- Mademoiselle (Mlle) es lo mismo que Miss y se referirá a mujeres solteras que no se han casado.
- Se separan los títulos por realeza (Jonkheer, Don, Sir, the Countess y Lady), "master" y oficiales (Captain, Colonel, Major, Dr y Rev).

```
data$Title <- as.character(data$Title)
royalty <- c("Jonkheer", "Don", "Sir", "the Countess", "Lady")
officer <- c("Capt", "Col", "Major", "Dr", "Rev")
data$Title[data$Title == "Mme"] <- "Mrs"
data$Title[data$Title == "Mlle"] <- "Miss"
data$Title[data$Title == "Ms"] <- "Mrs"
data$Title[data$Title %in% royalty] <- "Royalty"
data$Title[data$Title %in% officer] <- "Officer"
data$Title <- as.factor(data$Title)
```

Los títulos resultantes son:

```
table(data$Title)
```

```
##
## Master      Miss      Mr      Mrs Officer Royalty
##      40       184      517      127      18        5
```

Se comprueba que todos los registros con Mrs y Miss sean mujeres:

```
table(data$Sex[data$Title == "Mrs" | data$Title == "Miss"])
```

```
##
## female  male
##     311     0
```

También se comprueba que todos los registros con Mr sean hombres:

```
table(data$Sex[data$Title == "Mr"])
```

```
##  
## female    male  
##         0     517
```

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

En este dataset los valores vacíos se representan mediante el símbolo *NA*, además no hay valores extremos como 999 que den a entender valores perdidos.

A continuación, se muestra cada columna junto con el número de elementos vacíos que tiene.

```
mostrarCantidadCamposVacios <- function() {  
  sapply(data, function(x) sum(is.na(x)))  
}  
mostrarCantidadCamposVacios()  
  
## Survived    Pclass      Sex      Age      Fare      Cabin Embarked  
##         0         0         0      177      15      687         2  
## hasFamily   Title  
##         0         0
```

3.1.1. Age y Fare

Los datos perdidos de la edad representan el 19% de los datos. Se van a imputar los datos de los que no se dispone mediante el algoritmo KNN. Se ha elegido este algoritmo porque permite el uso de datos mixtos (continuos, nominales, etc.) para aproximar los valores faltantes y se parte de la hipótesis de que los registros de los pasajeros guardan cierta relación. El nuevo valor se imputará dependiendo de los *k* vecinos más cercanos.

```
emptyPercentage <- function(x){  
  sum(is.na(x))/nrow(data)  
}  
emptyPercentage(data$Age)  
  
## [1] 0.1986532
```

El problema que existe es que una persona solo puede tener de título “Master” si es igual o menor a 12 años. El algoritmo KNN no asegura que la predicción esté dentro de la regla anterior. Por este motivo, se van a separar los datos en dos partes: aquellos pasajeros que son “Master” y los que no.

```
isMaster <- data[data$Title == "Master",]
```

Se comprueba si hay pasajeros que son “Master” y el campo de la edad está vacío. Para estos pasajeros se va a elegir la media de la edad de las personas que tienen el título “Master” para rellenar los campos vacíos.

```
isMaster[is.na(isMaster$Age),]
```

```
##      Survived Pclass  Sex Age      Fare Cabin Embarked hasFamily  Title
## 66          1       3 male  NA 15.2458 <NA>      C          1 Master
## 160         0       3 male  NA 69.5500 <NA>      S          1 Master
## 177         0       3 male  NA 25.4667 <NA>      S          1 Master
## 710         1       3 male  NA 15.2458 <NA>      C          1 Master

meanAge <- mean(isMaster$Age[!is.na(isMaster$Age)])
data$Age[data$Title == "Master" & is.na(data$Age)] <- as.integer(meanAge)
```

Por último, se comprueba que no haya ningún pasajero mayor de 12 años con el título "Master".

```
data[data$Title == "Master" & data$Age > 12, ]

## [1] Survived  Pclass    Sex      Age      Fare      Cabin      Embark
## [8] hasFamily Title
## <0 rows> (or 0-length row.names)
```

Ahora se tratarán de imputar los valores para los pasajeros que no tengan el título "Master" utilizando todos los datos.

El algoritmo elegido utiliza distancias para aproximar el nuevo valor y como tiene que utilizar diferentes variables numéricas que no están en la misma escala se va a utilizar la normalización. Se normalizarán los atributos numéricos de "Fare" y "Edad". Finalmente, la edad y el coste del ticket se desnormalizarán para recuperar los valores iniciales.

```
normalizeData <- function(x, min, max){
  return ((x-min)/(max-min))
}

normalizedData <- data

agesWithoutNa <- data$Age[!is.na(data$Age)]
agesWithoutNa.max <- max(agesWithoutNa)
agesWithoutNa.min <- min(agesWithoutNa)

faresWithoutNa <- data$Fare[!is.na(data$Fare)]
faresWithoutNa.max <- max(faresWithoutNa)
faresWithoutNa.min <- min(faresWithoutNa)

normalizedData$Age[!is.na(normalizedData$Age)] <- normalizeData(agesWithoutNa, agesWithoutNa.min, agesWithoutNa.max)
normalizedData$Fare[!is.na(normalizedData$Fare)] <- normalizeData(faresWithoutNa, faresWithoutNa.min, faresWithoutNa.max)

normalizedData <- kNN(normalizedData, variable=c("Age"), k=3, imp_var = FALSE)
```



```
normalizedData <- kNN(normalizedData, variable=c("Fare"), k=3, imp_var = FALSE)

denormalize <- function(x,min,max) {
  return(x*(max-min) + min)
}

data$Age <- denormalize(normalizedData$Age, agesWithoutNa.min, agesWithoutNa.max)
data$Fare <- denormalize(normalizedData$Fare, faresWithoutNa.min, faresWithoutNa.max)
```

3.1.2. Cabin

Antes de decidir si imputar los valores perdidos de esta variable se comprobará cual es el porcentaje de valores vacíos respecto del total de los datos.

```
emptyPercentage(data$Cabin)

## [1] 0.7710438
```

Los valores perdidos representan el 77% de los datos. Debido al gran porcentaje que representan se va a optar por desechar este atributo.

```
data <- subset(data, select = -c(Cabin))
```

3.1.3. Embarked

Finalmente, solo quedan valores vacíos o perdidos en el atributo "Embarked". Estos valores representan el 0.002% de los datos totales, por lo tanto, el valor imputado será la moda o el valor más repetido.

```
emptyPercentage(data$Embarked)

## [1] 0.002244669
```

El valor más repetido es "S".

```
table(data$Embarked)

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

Se introduce el valor en los campos vacíos.

```
data$Embarked[is.na(data$Embarked)] <- "S"
```

La cantidad de campos vacíos debería de haber descendido a 0:

```
mostrarCantidadCamposVacios()
```

```
## Survived    Pclass      Sex      Age      Fare Embarked hasFamily
##          0         0         0         0         0         0         0
##      Title
##          0
```

3.2. Identificación y tratamiento de valores extremos

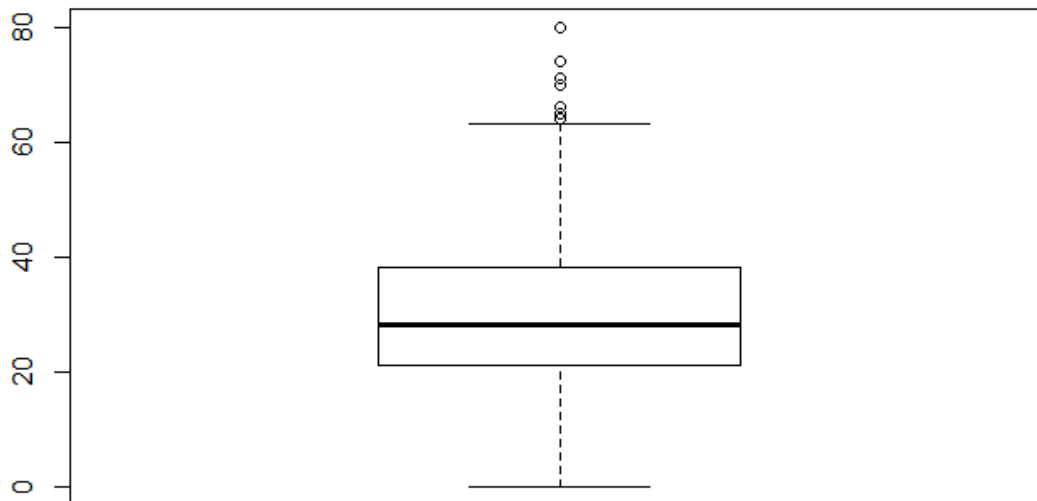
Los valores extremos o *outliers* son aquellos valores que se encuentran muy alejados de la distribución normal de una variable o población. Es decir, son observaciones que se desvían tanto del resto que levantan sospechas sobre si fueron generadas de la misma forma.

El primero paso será identificar dichos valores en los atributos numéricos “Age” y “Fare”.

3.2.1. Age

La variable “Age” según indica el siguiente *boxplot* cuenta con valores atípicos.

```
boxplot(data$Age)
```



Si solamente se muestran los valores atípicos:

```
levels(factor(boxplot.stats(data$Age)$out))
```

```
## [1] "64" "65" "66" "70" "71" "74" "80"
```

Los valores atípicos encontrados se ignorarán al ser valores que perfectamente pueden darse. El manejo de estos valores extremos consistirá en dejarlos como están.

3.2.2. Fare

Para obtener los valores extremos del precio del ticket se separan los datos según la clase social.

```
clase1 <- data[data$Pclass == "1",]
clase2 <- data[data$Pclass == "2",]
clase3 <- data[data$Pclass == "3",]
```

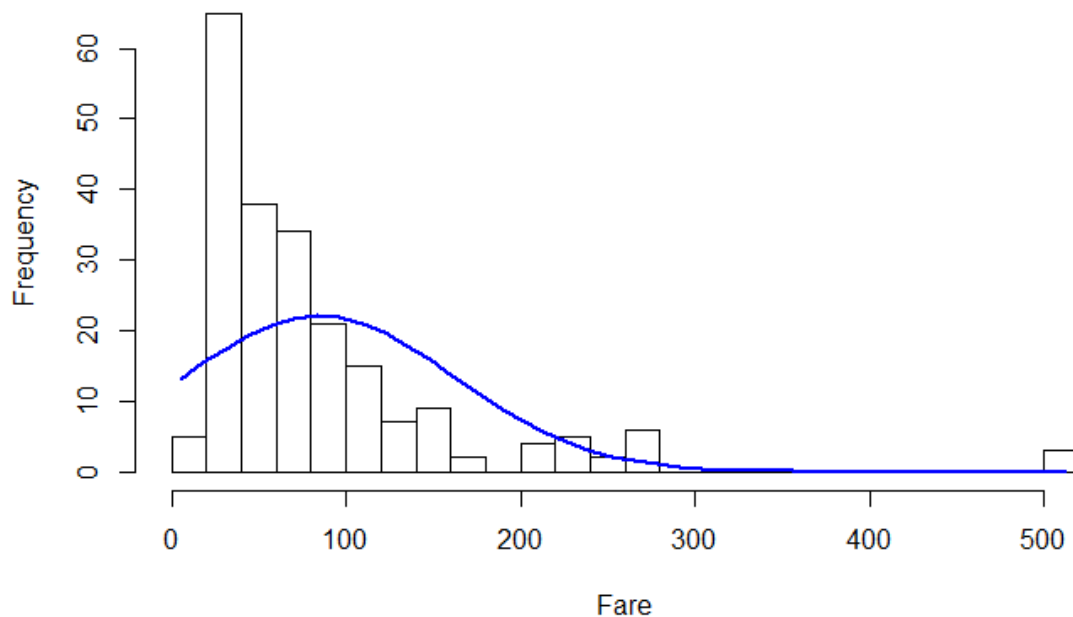
Los valores extremos de la clase 1 son:

```
levels(factor(boxplot.stats(clase1$Fare)$out))

## [1] "211.3375" "211.5"      "221.7792" "227.525"  "247.5208" "262.375"
## [7] "263"      "512.3292"
```

El histograma será:

```
showHistogram <- function(x, name, breaks){
  h <- hist(x, breaks=breaks, col="white", xlab=name, main="")
  xfit<-seq(min(x),max(x),length=40)
  yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
  yfit <- yfit*diff(h$mids[1:2])*length(x)
  lines(xfit, yfit, col="blue", lwd=2)
}
showHistogram(clase1$Fare, "Fare", 20)
```



En el histograma anterior se aprecia como los valores mayores de 200 son menos frecuentes, pero parecen valores válidos para precios de billetes de primera clase. Sin embargo, el valor 512 sí que parece ser un valor atípico. Se sustituye por NA.

```
data$Fare[data$Pclass == "1" & data$Fare == "512.3292"] <- NA
```

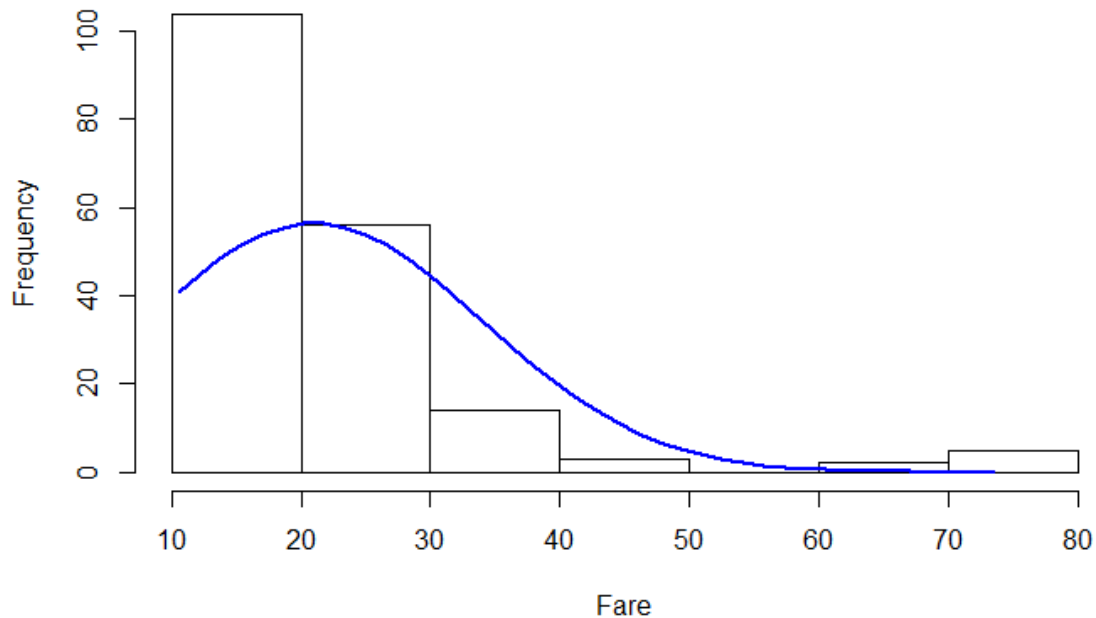
Los valores extremos de la clase 2 son:

```
levels(factor(boxplot.stats(clase2$Fare)$out))

## [1] "65"      "73.5"
```

Aunque estos valores anteriores se han detectado como valores extremos son valores que pueden ser válidos porque no hay mucha diferencia entre estos y los valores más frecuentes que se dan en el precio de los tickets de esta clase socio-económica. Puede que sean más caros por ser tickets comprados a última hora. De momento estos valores se mantienen.

```
showHistogram(clase2$Fare, "Fare", 8)
```



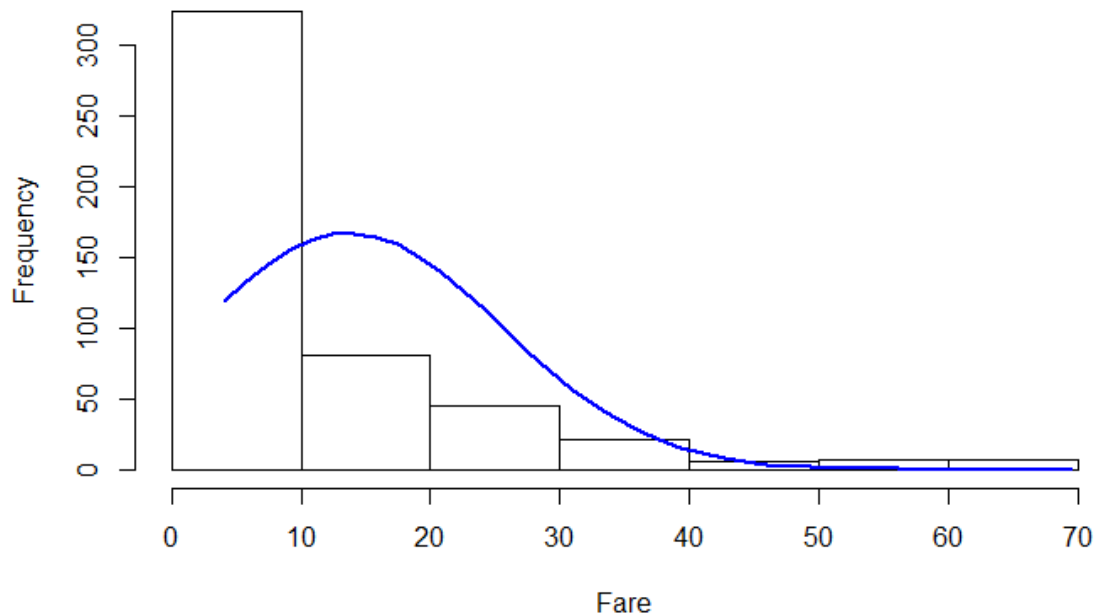
Los valores extremos de la clase 3 son:

```
levels(factor(boxplot.stats(clase3$Fare)$out))
```

```
## [1] "27.9"    "29.125"  "31.275"  "31.3875" "34.375"  "39.6875" "46.9"  
## [8] "56.4958" "69.55"
```

Igual que en caso anterior los valores detectados como *outliers* no son tan diferentes del resto. Por este motivo, estos valores se van a mantener teniendo presente que igual son tickets comprados a última hora.

```
showHistogram(clase3$Fare, "Fare", 5)
```



3.2.3. Imputación outliers

Para imputar los valores faltantes del atributo “Fare” se utilizará de nuevo el algoritmo KNN y la normalización de la edad y del precio de los tickets.

```
normalizedData <- data

faresWithoutNa <- data$Fare[!is.na(data$Fare)]
faresWithoutNa.max <- max(faresWithoutNa)
faresWithoutNa.min <- min(faresWithoutNa)

normalizedData$Age <- normalizeData(normalizedData$Age, min(normalizedData$Age), max(normalizedData$Age))
normalizedData$Fare[!is.na(normalizedData$Fare)] <- normalizeData(faresWithoutNa, faresWithoutNa.min, faresWithoutNa.max)

normalizedData <- kNN(normalizedData, variable=c("Fare"), k=3, imp_var = FALSE)

data$Fare <- denormalize(normalizedData$Fare, faresWithoutNa.min, faresWithoutNa.max)
```

3.3. Exportación de datos

Una vez que se han procesado y tratado los datos se procederá a guardar estos en un nuevo fichero llamado “train_clean.csv”.

```
write.csv(data, "data/train_clean.csv")
```

4. Análisis de los datos

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

Mediante el análisis de los datos se intentará responder a preguntas como:

- ¿Es la media de edad de los que se han salvado menor a los que han fallecido?
- ¿Qué relación hay entre las diferentes características de los pasajeros y el atributo que indica si han sobrevivido?

Para contestar a este tipo de preguntas se seleccionan varios grupos de dentro del conjunto de datos que pueden resultar interesantes para analizar y/o comparar. No todos los grupos aquí seleccionados se usarán en las pruebas estadísticas.

```
# Agrupación de Los que se han salvado y de Los que no
survived <- data[data$Survived == 1,]
notSurvived <- data[data$Survived == 0,]

# Agrupación por sexo
female <- data[data$Sex == "female",]
male <- data[data$Sex == "male",]

# Agrupación por si tienen familia
hasFamily <- data[data$hasFamily == 1,]
notFamily <- data[data$hasFamily == 0,]
```

4.2. Comprobación de la normalidad y homogeneidad de la varianza

Para comprobar si los valores que toman las variables cuantitativas de edad “Age” y el precio del ticket “Fare” provienen de una población normal, se utilizará el test de *Shapiro-Wilk*. Este test se considera uno de los métodos más potentes para contrastar la normalidad. El nivel de significación elegido es 0.05.

Se parte de la hipótesis nula de que la población está distribuida normalmente. Si el p-valor es menor al nivel de significación elegido entonces la hipótesis de que la población es normal se rechaza y se concluye que dichos datos no cuentan con una distribución normal.

```
shapiro.test(data$Age)

##
##  Shapiro-Wilk normality test
##
## data:  data$Age
## W = 0.97698, p-value = 1.185e-10

shapiro.test(data$Fare)

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

```
##  
## data: data$Fare  
## W = 0.59908, p-value < 2.2e-16
```

En ambos casos el p-valor es menor a 0.05, por lo que se acepta la hipótesis de que la distribución de estas variables no es normal.

Si las muestras tienen más de 30 datos se puede aplicar el *teorema del límite central* y asumir que la distribución de la media es una normal. Se comprueba el tamaño de la muestra de los pasajeros que han sobrevivido y los que no.

Se podrán usar test paramétricos para la comparación de la media ya que la distribución de la media es una normal, aunque la distribución de los datos no sea normal.

```
nrow(survived)  
## [1] 342  
nrow(notSurvived)  
## [1] 549
```

Seguidamente, se estudia la homogeneidad de varianzas mediante el test no paramétrico de *Fligner-Killeen*. En este caso, se estudia la homogeneidad de la varianza de la edad entre grupos que han sobrevivido y los que no.

```
fligner.test(Age ~ Survived, data)  
  
##  
## Fligner-Killeen test of homogeneity of variances  
##  
## data: Age by Survived  
## Fligner-Killeen:med chi-squared = 1.2029e-05, df = 1, p-value =  
## 0.9972
```

Como el p-valor es mayor al nivel de significación (0.05) elegido se acepta la hipótesis nula de que las varianzas de ambas muestras son homogéneas.

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

4.3.1. ¿Es la edad de las personas que han sobrevivido menor a las que han fallecido?

La primera prueba estadística que se aplicará consistirá en un contraste de hipótesis de dos muestras independientes sobre la media para determinar si la media de la edad de las personas que han sobrevivido es menor a las que han fallecido.

La primera muestra tendrá las edades de aquellos pasajeros que han sobrevivido y la otra muestra tendrá las edades restantes.

Como lo que se quiere comparar es la media y anteriormente se ha establecido que la distribución muestral de la media de estos dos grupos es una normal gracias al

teorema del límite central, se aplicará un test paramétrico unilateral donde la varianza poblacional es desconocida, pero asumiendo homogeneidad (como se ha demostrado en el apartado anterior).

Las hipótesis serán:

$$H_0: \mu_s = \mu_{ns}$$

$$H_1: \mu_s < \mu_{ns}$$

El nivel de significación elegido es 0.05.

```
t.test(survived$Age, notSurvived$Age, alternative = "less", var.equal = TRUE)

##
## Two Sample t-test
##
## data: survived$Age and notSurvived$Age
## t = -3.0881, df = 889, p-value = 0.001039
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -1.456614
## sample estimates:
## mean of x mean of y
## 27.92690 31.04736
```

Puesto que 0.001039 es menor que el nivel de significación, entonces se rechaza la hipótesis nula en favor de la hipótesis alternativa. La media de la edad de las personas que sobrevivieron es menor a la media de la edad de los que no sobrevivieron. Se puede deducir que se salvaron más jóvenes que gente mayor.

4.3.2. Correlación entre la supervivencia y el resto de atributos

La variable que indica si un pasajero ha sobrevivido es una variable binaria, y no todos los atributos restantes son de un mismo tipo, por lo que se usarán diferentes técnicas y métodos para decidir la correlación entre los atributos.

Survived y variables categóricas (Sex, Embarked, Pclass, hasFamily y Title)

Para comprobar si existe relación entre la variable “Survived” y las variables categóricas de los datos se usará el test *Chi-Square Independence*.

Este test usará dos hipótesis:

H_0 : x es independiente de y

H_1 : x no es independiente de y


```

chiSquareTest <- function(){
  columns <- c("Sex", "Embarked", "Pclass", "hasFamily", "Title")
  corr_matrix <- matrix(nc=1, nr=0)
  colnames(corr_matrix) <- c("p-value")
  for (columnName in columns) {
    contingencyTable <- table(data$Survived, data[,columnName])
    testResult <- chisq.test(contingencyTable, correct = FALSE)

    pair = matrix(ncol=1, nrow=1)
    pair[1] = testResult$p.value
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- columnName
  }
  print(corr_matrix)
}

chiSquareTest()

##                p-value
## Sex            3.711748e-59
## Embarked       2.300863e-06
## Pclass         4.549252e-23
## hasFamily      1.275675e-09
## Title          1.390148e-60

```

Como todos los p-valores de las variables categóricas son menores que el nivel de significación 0.05 se rechaza la hipótesis nula de que las variables “Sex”, “Embarked”, “Pclass”, “hasFamily” y “Title” son independientes de “Survived”. Es decir, este test demuestra que hay cierta relación entre estas variables y si una persona sobrevivió.

Survived y variables numéricas (Edad y Fare)

Para calcular la correlación entre una variable categórica de dos niveles y una numérica se puede usar la regresión lineal generando un modelo para cada par Survived-Age y Survived-Fare.

No se ha utilizado el test paramétrico *One-Way ANOVA* ya que este requiere que las variables sigan una distribución normal, y este no es el caso de la edad y del precio del ticket.

En este caso la variable independiente será “Survived” y las variables dependientes serán “Age” y “Fare”.

```

model <- lm(formula = Age ~ Survived, data = data)
summary(model)

##
## Call:
## lm(formula = Age ~ Survived, data = data)
##
## Residuals:

```

```
##      Min      1Q  Median      3Q      Max
## -30.047 -10.047  -1.927   8.073  52.073
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   31.047      0.626  49.593 < 2e-16 ***
## Survived1     -3.120      1.010  -3.088 0.00208 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.67 on 889 degrees of freedom
## Multiple R-squared:  0.01061,    Adjusted R-squared:  0.0095
## F-statistic: 9.536 on 1 and 889 DF,  p-value: 0.002077
```

A partir de los coeficientes se sabe que la ecuación del modelo queda:

$$Age = 31.047 - 3.120 * Survived$$

La variable Survived será 1 cuando haya sobrevivido y 0 cuando no, por este motivo solo aparece “Survived1” en la lista de coeficientes del modelo. Cada coeficiente aparece junto a su error estándar y acompañado del valor de un estadístico t de Student y un p-valor que permiten contrastar la significación del parámetro en cuestión y resolver los siguientes contrastes de hipótesis:

$H_0: B_i = 0$ La variable X_i no es explicativa

$H_1: B_i \neq 0$ La variable X_i es explicativa

Mediante estos contrastes se pretende determinar si una variable X_i figura en el modelo de regresión y, por tanto, si es o no una variable explicativa del comportamiento de la variable Y.

En esta ocasión el p-valor de la variable que indica que el pasajero sobrevive es menor al nivel de significación del 0.05. De esta manera, la hipótesis nula se rechaza concluyendo que la variable “Survived” es explicativa de la variable “Age”.

El coeficiente de determinación (R^2) ajustado es 0.0095. Se puede aventurar que es bastante malo al acercarse al cero. El modelo de regresión explica un 0.095% de la variabilidad total de las observaciones.

Este modelo es muy simple, pero a través de él se puede reafirmar lo que ya se sabe; la edad de una persona que sobrevive (el coeficiente es negativo) será menor que la de una persona que no ha sobrevivido.

```
model <- lm(formula = Fare ~ Survived, data = data)
summary(model)

##
## Call:
```

```
## lm(formula = Fare ~ Survived, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -37.689 -15.324 -13.751   5.336 240.586
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.414      1.697   13.205 < 2e-16 ***
## Survived1     22.251      2.740    8.122 1.53e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.77 on 889 degrees of freedom
## Multiple R-squared:  0.06908,    Adjusted R-squared:  0.06803
## F-statistic: 65.96 on 1 and 889 DF,  p-value: 1.527e-15
```

Según el modelo regresión este atributo es más significativo estadísticamente que la edad al ser el p-valor mucho menor. Además, la calidad del modelo, aunque es mala ha mejorado bastante y explica un 6.803% de la variabilidad total de las observaciones.

Según la salida del modelo, alguien que ha sobrevivido ha pagado 22.25 más por el billete que alguien que no ha sobrevivido.

El atributo “Fare” será uno que habrá que tener en cuenta.

4.3.2. Modelo supervisado

Se va a entrenar un modelo supervisado para predecir si un pasajero sobrevive. Antes de elegir modelo se van a separar los datos mediante el método *holdout*. El 70% de los datos serán de entrenamiento “train” y el 30% serán para comprobar el *accuracy* del modelo “test”. La partición de los datos será estratificada manteniendo así el mismo porcentaje de instancias de cada valor de la clase (Survived: 0 o 1).

```
h <- holdout(data$Survived, ratio=0.7, mode="stratified")
dataTrain <- data[h$tr,]
dataTest <- data[h$ts,]
```

El modelo elegido es *Random Forest*. Se ha elegido este modelo como modelo base (“baseline”) al ser bastante popular. Este modelo recibe dos argumentos: *mtry* será el número máximo de variables en los árboles creados y *ntree* será el número de árboles a realizar.

```
printResults <- function(model){
  pred <- predict(model, dataTest, OOB=TRUE, type="class")
  print(confusionMatrix(dataTest$Survived, pred)$overall[ 'Accuracy' ])
}
```

Se prueban varios modelos con distintos atributos:

```

# Todos Los atributos
model1 <- randomForest(Survived ~ Title + Sex + Pclass + hasFamily + Embarked + Age + Fare, data=dataTrain, ntree=500)
printResults(model1)

## Accuracy
## 0.8320896

# Solo el título, el sexo y La clase del pasajero
model2 <- randomForest(Survived ~ Title + Sex + Pclass, data=dataTrain, ntree=500)
printResults(model2)

## Accuracy
## 0.7835821

# Solo el título, el sexo, La clase y si un pasajero tiene familia
model3 <- randomForest(Survived ~ Title + Sex + Pclass + hasFamily, data=dataTrain, ntree=500)
printResults(model3)

## Accuracy
## 0.8059701

# Solo el sexo, La clase, si una persona tiene familia y La edad
model4 <- randomForest(Survived ~ Sex + Fare + hasFamily + Age, data=dataTrain, ntree=500)
printResults(model4)

## Accuracy
## 0.8208955

```

El mejor modelo *baseline* conseguido tiene un 0.83 de *accuracy* y solo utiliza los atributos: sexo, el precio del ticket, si una persona tiene familia y la edad del pasajero.

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

A continuación, se procederá a visualizar los análisis anteriores de forma visual mediante gráficos.

5.1. Normalidad

Mediante los histogramas siguientes y los gráficos Q-Q se demuestra claramente que las variables numéricas de estos datos no siguen una distribución normal. Esto se ha comprobado anteriormente mediante el test de *Shapiro-Wilk*.

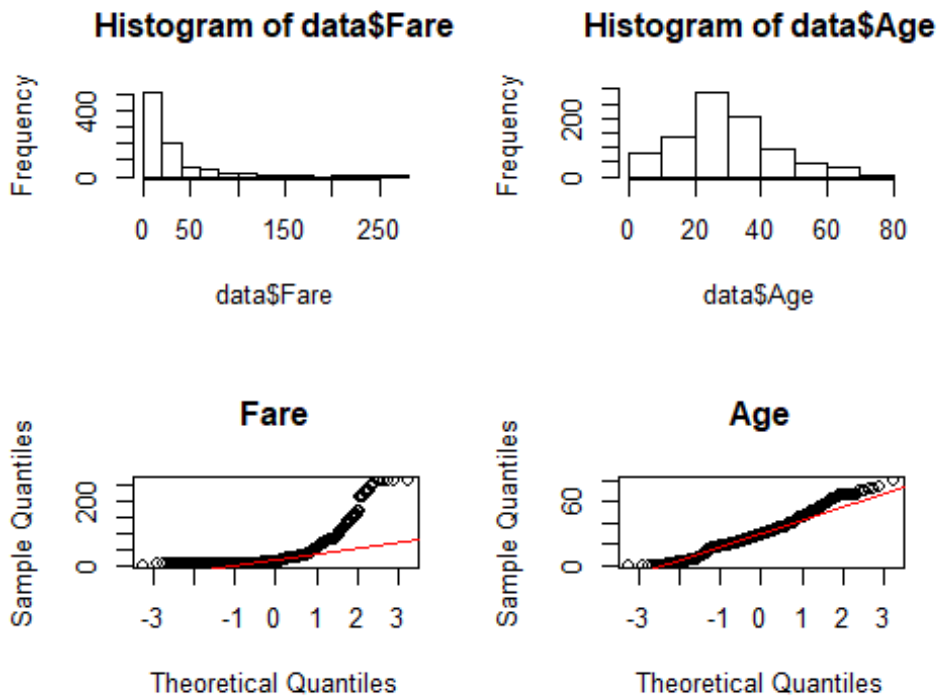
```

par(mfrow=c(2,2))

hist(data$Fare)
hist(data$Age)

```

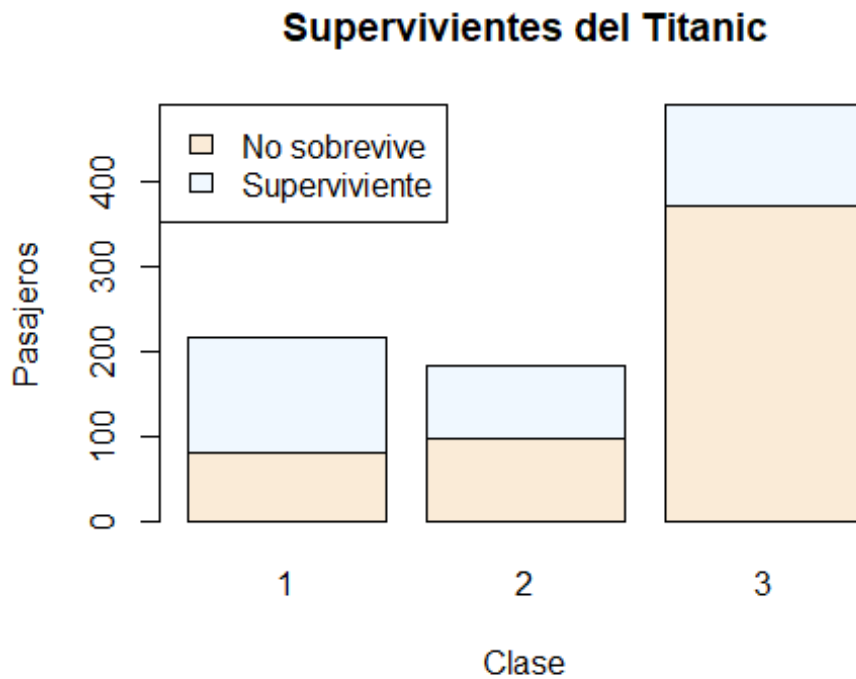
```
qqnorm(data$Fare, main="Fare")
qqline(data$Fare,col=2)
qqnorm(data$Age, main="Age")
qqline(data$Age,col=2)
```



5.2. Supervivientes del Titanic por clase social

Mediante el gráfico siguiente se observa que sobreviven más personas de la primera clase. La clase que más fallecidos tiene es la tercera.

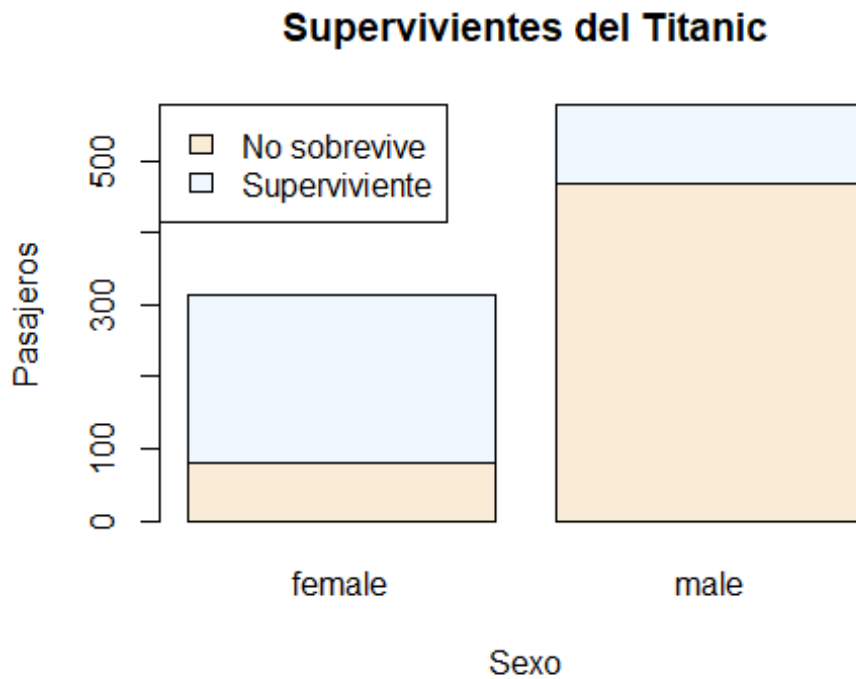
```
titanic.data<-table(data[,c(1,2)])
barplot(titanic.data, main = "Supervivientes del Titanic", xlab = "Clase",
,col= c("antiquewhite","aliceblue"), ylab= "Pasajeros")
legend("topleft", c("No sobrevive","Superviviente"), fill =c("antiquewhite",
"aliceblue"))
```



5.3. Supervivientes del Titanic por sexo

Se ve claramente como han sobrevivido más mujeres que hombres, y que una gran parte de los pasajeros que eran hombres no han sobrevivido. Existe una correlación entre las dos variables, como se ha probado en apartados anteriores.

```
titanic.data<-table(data[,c(1,3)])  
barplot(titanic.data, main = "Supervivientes del Titanic", xlab = "Sexo",  
col= c("antiquewhite","aliceblue"), ylab="Pasajeros")  
legend("topleft", c("No sobrevive","Superviviente"), fill =c("antiquewhite",  
"aliceblue"))
```

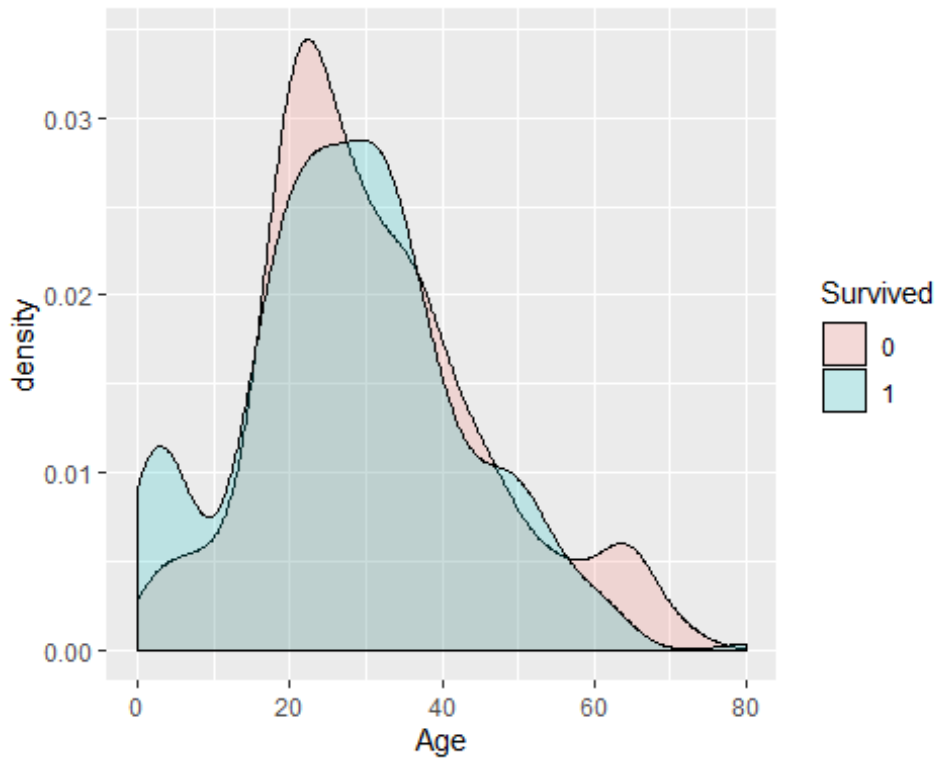


5.4. Supevivientes del Titanic por edad

Se confirma la hipótesis aceptada de que la media de edad de las personas que han sobrevivido es menor que la de los pasajeros que no han sobrevivido. Se observa que hay un pico hasta los cinco años, comprobando así que hubo más supervivientes que fallecidos entre los niños y niñas de corta edad.

Aproximadamente, a partir de los años 57 hay menos supervivientes que los fallecidos.

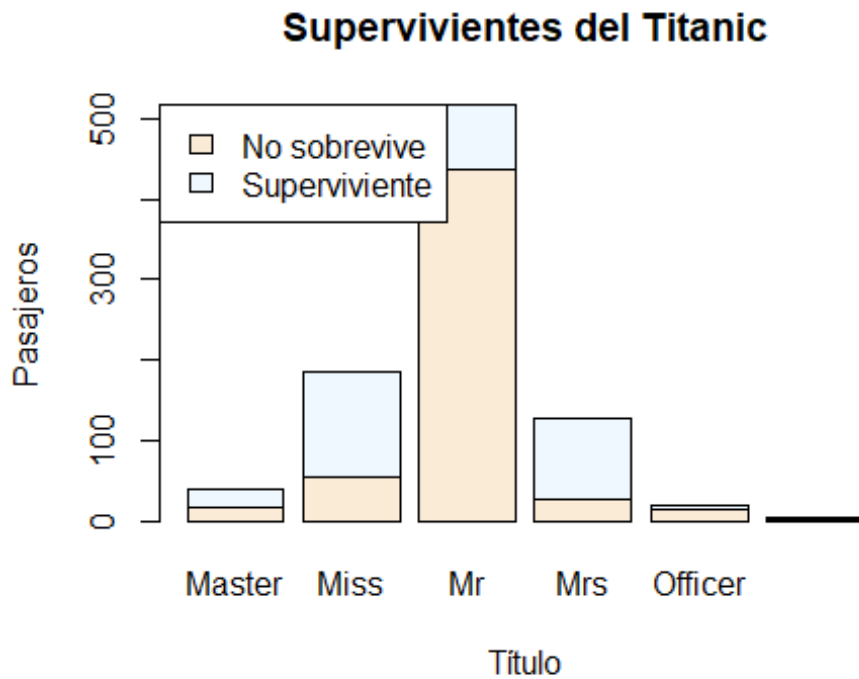
```
ggplot(data, aes(Age, fill = Survived)) + geom_density(alpha = 0.2)
```



5.5. Supervivientes del Titanic por título

Las personas que más se salvaron fueron las mujeres, es decir, pasajeros con título "Miss" y "Mrs".

```
titanic.data<-table(data[,c(1,8)])
barplot(titanic.data, main = "Supervivientes del Titanic", xlab = "Título",
,col= c("antiquewhite","aliceblue"), ylab="Pasajeros")
legend("topleft", c("No sobrevive","Superviviente"), fill =c("antiquewhite",
"aliceblue"))
```

6. Conclusiones

Se han realizado tres tipos de pruebas estadísticas sobre un conjunto de datos que se correspondía con diferentes atributos relativos a los pasajeros del Titanic con motivo de poder responder a la pregunta u objetivo que se planteaba al comienzo del informe.

Mediante el contraste de hipótesis realizado se ha podido comprobar como la edad media de las personas que han sobrevivido es menor a la de aquellos pasajeros que han fallecido. La conclusión obtenida mediante este contraste de hipótesis se puede observar de forma visual en los gráficos realizados en el apartado anterior.

Por otro lado, el análisis de correlación realizado ha permitido conocer cuáles son las variables que están más relacionadas con la supervivencia de un pasajero. Una vez conocidas estas, se ha procedido a implementar varios modelos supervisados dividiendo los datos según el método *holdout*. Se ha elegido el modelo que ha tenido el mayor porcentaje en la métrica *Accuracy* como el modelo a utilizar para predecir si un pasajero sobrevivió o no.

Además, previamente los datos se han limpiado evitando inconsistencias, casos de ceros o elementos vacíos y valores extremos (también llamados *outliers*). Los valores faltantes se han imputado utilizando algoritmos como bien puede ser el KNN y/o eligiendo la moda/media de ciertos valores.

7. Recursos

1. Subirats, L., Oswaldo, D., & Calvo, M. (2019). *Introducción a la limpieza y análisis de los datos*. UOC.
2. Gibergans, J. *Contraste de dos muestras*. UOC.
3. Gibergans, J. *Regresión lineal simple*. UOC.
4. Osborne, J. (2013). *Best Practices in Data Cleaning*. SAGE Publications, Inc.