Práctica 2: Dataset Titanic

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El hundimiento del Titanic es uno de los naufragios más infames de la historia.

El 15 de abril de 1912, durante su viaje inaugural, el ampliamente considerado "insumergible" RMS Titanic se hundió después de chocar con un iceberg. Desafortunadamente, no había suficientes botes salvavidas para todos a bordo, resultando en la muerte de 1502 de 2224 pasajeros y tripulación.

Aunque había un elemento de suerte en la supervivencia, parece que algunos grupos de personas tenían más probabilidades de sobrevivir que otros.

En este desafío, les pedimos que construyan un modelo predictivo que responda a la pregunta: "¿qué tipo de personas tenían más probabilidades de sobrevivir?" usando los datos de los pasajeros (es decir, nombre, edad, sexo, clase socioeconómica, etc.).

usar machine learning para crear un modelo que prediga qué pasajeros sobrevivieron al naufragio del Titanic.

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

La integración (combinación de datos de distintas fuentes), selección (filtrado de los datos de interés) y reducción (representación reducida de los datos manteniendo la integridad de la muestra original) corresponden a la limpieza de los datos.

Los datos de los que disponemos son: Train.csv: contiene los detalles () de un subconjunto de los pasajeros a bordo (891 para ser exactos) y, además revela si sobrevivieron o no. Test.csv: contiene información similar al train.csv pero no revela si cada pasajero sobrevivió o no al desastre.

Si no queremos hacer predicciones sobre la supervivencia de los pasajeros del test.csv podemos integrar los dos datasets para hacer otro tipo de tests estadísticos.

```
train <- read.csv('C:/Users/ester/Desktop/Kaggle/train.csv')
test <- read.csv('C:/Users/ester/Desktop/Kaggle/test.csv')
datos <- rbind(train[,-2], test)
```

3. Limpieza de los datos.

Además creamos alguna nueva variable que puede ser interesante para los estudios estadísticos posteriores.

```
head(train)
```

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

```
str(train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley
(Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques
Heath (Lily May Peel)" ...
##  $ 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
train$Pclass <- factor(train$Pclass, levels = c(1,2,3), labels = c('First', 'Second', 'Third'))
train$Sex <- factor(train$Sex, levels = c('male','female'), labels = c('Male', 'Female'))
train$Embarked <- factor(train$Embarked, levels = c('C','Q','S'), labels = c('Cherbourg', 'Queenstown', 'Southampton'))
str(train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : Factor w/ 3 levels "First","Second",..: 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
##  $ Sex        : Factor w/ 2 levels "Male","Female": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 3 levels "Cherbourg","Queenstown",..: 3 1 3 3 3 2 3 3 3 1 ...
```

```r
summary(train)
```

```
##   PassengerId       Survived         Pclass        Name               Sex
##  Min.   :  1.0   Min.   :0.0000   First :216   Length:891         Male  :577
##  1st Qu.:223.5   1st Qu.:0.0000   Second:184   Class :character   Female:314
##  Median :446.0   Median :0.0000   Third :491   Mode  :character
##  Mean   :446.0   Mean   :0.3838
##  3rd Qu.:668.5   3rd Qu.:1.0000
##  Max.   :891.0   Max.   :1.0000
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare           Cabin                 Embarked
##  Min.   :  0.00   Length:891         Cherbourg  :168
```

```
##  1st Qu.:  7.91   Class :character   Queenstown : 77
##  Median : 14.45   Mode  :character   Southampton:644
##  Mean   : 32.20                      NA's       :  2
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

```
strsplit(train$Name, split = ' ')[2]
```

```
## [[1]]
## [1] "Cumings,"  "Mrs."      "John"      "Bradley"   "(Florence" "Briggs"
## [7] "Thayer)"
```

```
formula <- unlist(sapply(strsplit(train$Name, ", "), function(x) x[2],
simplify=FALSE))
train$formula1 <- unlist(sapply(strsplit(formula, ". "), function(x) x[1],
simplify=FALSE))
train$Family_Name <- unlist(sapply(strsplit(train$Name, ". "), function(x)
x[1], simplify=FALSE))
```

```
head(test)
```

```
##   PassengerId Pclass                                          Name    Sex
Age
## 1         892      3                              Kelly, Mr. James   male
34.5
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female
47.0
## 3         894      2                     Myles, Mr. Thomas Francis   male
62.0
## 4         895      3                              Wirz, Mr. Albert   male
27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female
22.0
## 6         897      3                    Svensson, Mr. Johan Cervin   male
14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked
## 1     0     0  330911  7.8292              Q
## 2     1     0  363272  7.0000              S
## 3     0     0  240276  9.6875              Q
## 4     0     0  315154  8.6625              S
## 5     1     1 3101298 12.2875              S
## 6     0     0    7538  9.2250              S
```

```
str(test)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)"
"Myles, Mr. Thomas Francis" "Wirz, Mr. Albert" ...
```

```
##  $ Sex         : chr  "male" "female" "male" "male" ...
##  $ Age         : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp       : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch       : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket      : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare        : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin       : chr  "" "" "" "" ...
##  $ Embarked    : chr  "Q" "S" "Q" "S" ...
```

```
test$Pclass <- factor(test$Pclass, levels = c(1,2,3), labels = c('First',
'Second', 'Third'))
test$Sex <- factor(test$Sex, levels = c('male','female'), labels = c('Male',
'Female'))
test$Embarked <- factor(test$Embarked, levels = c('C','Q','S'), labels =
c('Cherbourg', 'Queenstown', 'Southampton'))
str(test)
```

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

```
summary(test)
```

```
##   PassengerId        Pclass        Name               Sex           Age
##  Min.    : 892.0   First :107   Length:418          Male   :266   Min.   :
0.17
##   1st Qu.: 996.2    Second: 93    Class :character    Female:152    1st
Qu.:21.00
##  Median :1100.5    Third :218    Mode  :character                  Median
:27.00
##  Mean    :1100.5                                                    Mean
:30.27
##  3rd Qu.:1204.8                                                     3rd
Qu.:39.00
##  Max.    :1309.0                                                    Max.
:76.00
##                                                                     NA's   :86
##      SibSp            Parch           Ticket               Fare
```

```
##  Min.   :0.0000   Min.   :0.0000   Length:418         Min.   :  0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   Class :character   1st Qu.:  7.896
##  Median :0.0000   Median :0.0000   Mode  :character   Median : 14.454
##  Mean   :0.4474   Mean   :0.3923                      Mean   : 35.627
##  3rd Qu.:1.0000   3rd Qu.:0.0000                      3rd Qu.: 31.500
##  Max.   :8.0000   Max.   :9.0000                      Max.   :512.329
##                                                       NA's   :1
##     Cabin                  Embarked
##  Length:418         Cherbourg  :102
##  Class :character   Queenstown : 46
##  Mode  :character   Southampton:270
##
##
##
##
```

```
strsplit(test$Name, split = ' ')[2]
```

```
## [[1]]
## [1] "Wilkes," "Mrs."    "James"   "(Ellen"  "Needs)"
```

```
formula <- unlist(sapply(strsplit(test$Name, ", "), function(x) x[2], simplify=FALSE))
test$formula1 <- unlist(sapply(strsplit(formula, ". "), function(x) x[1], simplify=FALSE))
test$Family_Name <- unlist(sapply(strsplit(test$Name, ". "), function(x) x[1], simplify=FALSE))
```

```
head(datos)
```

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

```r
str(datos)
```

```
## 'data.frame':    1309 obs. of  11 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ 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)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques
Heath (Lily May Peel)" ...
##  $ 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
datos$Pclass <- factor(datos$Pclass, levels = c(1,2,3), labels = c('First',
'Second', 'Third'))
datos$Sex  <-  factor(datos$Sex,  levels  =  c('male','female'),  labels  =
c('Male', 'Female'))
datos$Embarked <- factor(datos$Embarked, levels = c('C','Q','S'), labels =
c('Cherbourg', 'Queenstown', 'Southampton'))
str(datos)
```

```
## 'data.frame':    1309 obs. of  11 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Pclass     : Factor w/ 3 levels "First","Second",..: 3 1 3 1 3 3 1 3 3
2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley
(Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques
Heath (Lily May Peel)" ...
##  $ Sex        : Factor w/ 2 levels "Male","Female": 1 2 2 2 1 1 1 1 2 2
...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803"
...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 3 levels "Cherbourg","Queenstown",..: 3 1 3 3 3
2 3 3 3 1 ...
```

```r
summary(datos)
```

```
##   PassengerId      Pclass        Name              Sex           Age
##  Min.   :   1   First :323   Length:1309        Male  :843   Min.   : 0.17
##  1st Qu.: 328   Second:277   Class :character   Female:466   1st Qu.:21.00
##  Median : 655   Third :709   Mode  :character                Median :28.00
```

```
##  Mean   : 655                                                    Mean   :29.88
##  3rd Qu.: 982                                                    3rd Qu.:39.00
##  Max.   :1309                                                    Max.   :80.00
##                                                                  NA's   :263
##      SibSp            Parch          Ticket              Fare
##  Min.   :0.0000   Min.   :0.000   Length:1309        Min.   :  0.000
##  1st Qu.:0.0000   1st Qu.:0.000   Class :character   1st Qu.:  7.896
##  Median :0.0000   Median :0.000   Mode  :character   Median : 14.454
##  Mean   :0.4989   Mean   :0.385                      Mean   : 33.295
##  3rd Qu.:1.0000   3rd Qu.:0.000                      3rd Qu.: 31.275
##  Max.   :8.0000   Max.   :9.000                      Max.   :512.329
##                                                      NA's   :1
##     Cabin                    Embarked
##  Length:1309        Cherbourg  :270
##  Class :character   Queenstown :123
##  Mode  :character   Southampton:914
##                     NA's       :  2
##
##
##
```

```
strsplit(datos$Name, split = ' ')[2]
```

```
## [[1]]
## [1] "Cumings,"  "Mrs."      "John"      "Bradley"   "(Florence" "Briggs"
## [7] "Thayer)"
```

```
formula <- unlist(sapply(strsplit(datos$Name, ", "), function(x) x[2],
simplify=FALSE))
datos$formula1 <- unlist(sapply(strsplit(formula, ". "), function(x) x[1],
simplify=FALSE))
datos$Family_Name <- unlist(sapply(strsplit(datos$Name, ". "), function(x)
x[1], simplify=FALSE))
```

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

Vemos que algunas variables contienen ceros y/o elementos vacíos.

Las variables `SibSp`, `Parch` y `Fare` contienen ceros.

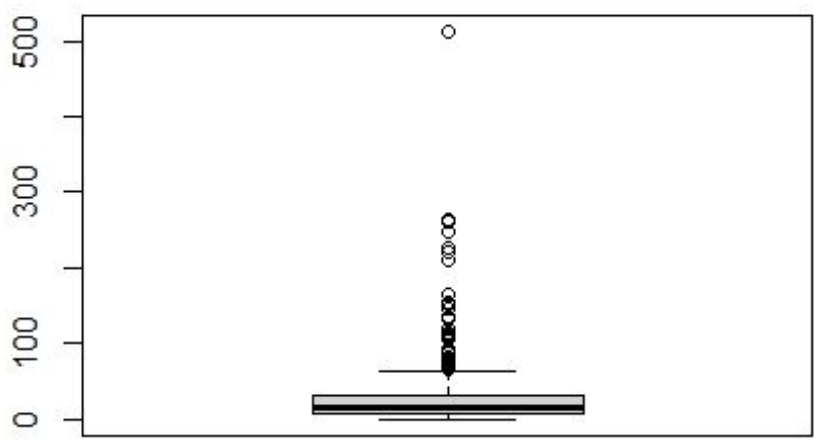
La variable `SibSp` hace referencia al número de hermanos/conyuges a bordo del Titanic, con que los ceros tienen sentido y entran dentro del rango de la variable.

La variable `Pacrh` hace referencia al número de padre e hijos a bordo del Titanic, con lo cual los ceros tmb tienen sentido y entran dentro del rango de valores admisibles para la variable.

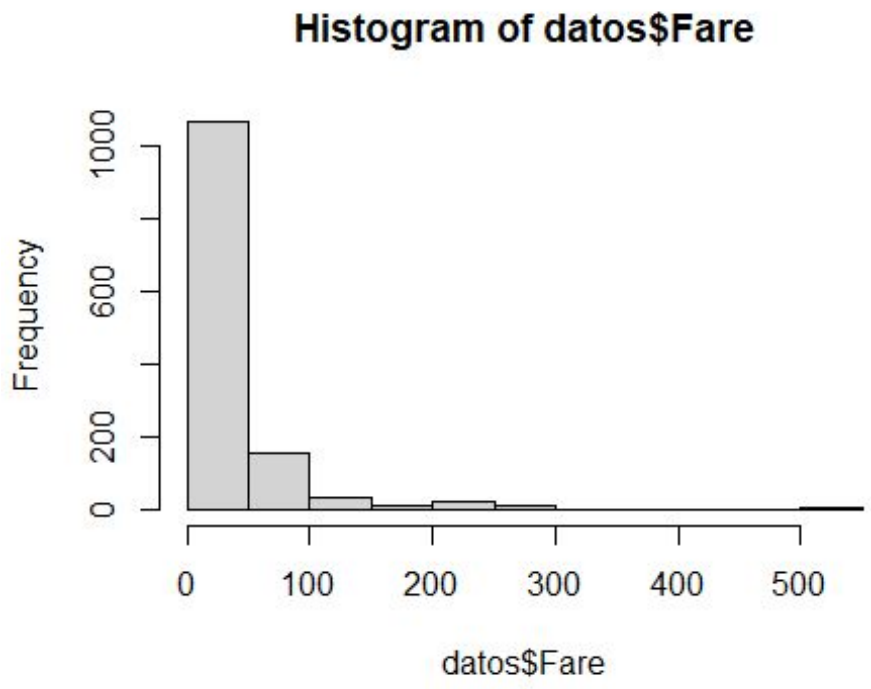
La variable `Fare` hace referencia a la tarifa que pagaron los pasajeros por su ticket. Aparecen 17 valores 0, no sabemos si es un error o esos pasajeros viajaron gratis.

Entre la gente que que viajaba en el Titanic había tripulacion y pasajeros, podemos suponer que esos 17 0 que aparecen son debidos a la tripulación que aparece en el dataset.

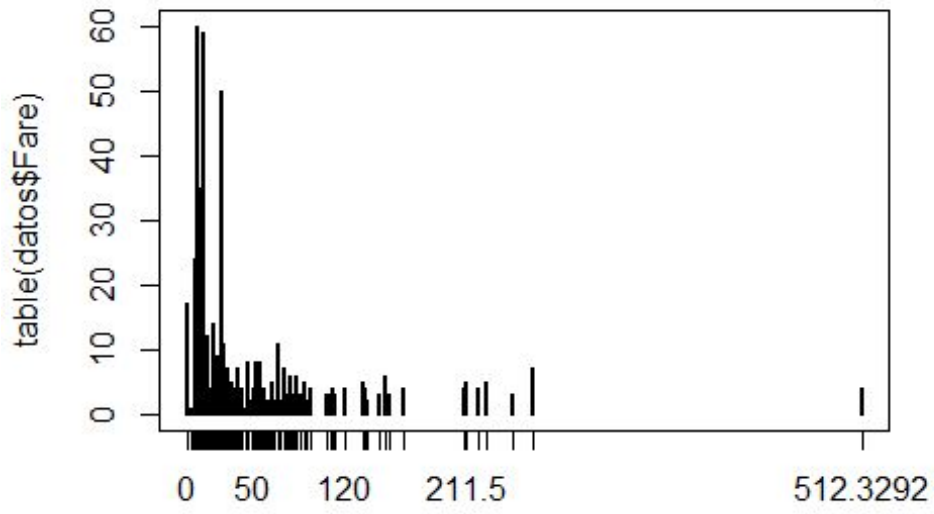
```
boxplot(datos$Fare)
```




```
hist(datos$Fare)
```




```
plot(table(datos$Fare))
```



```
datos_0 <- datos[datos$Fare == 0,]
head(datos_0)
```

```
##      PassengerId Pclass                                Name  Sex Age SibSp
Parch
## 180          180  Third               Leonard, Mr. Lionel Male  36     0
0
## 264          264  First             Harrison, Mr. William Male  40     0
0
## 272          272  Third     Tornquist, Mr. William Henry Male  25     0
0
## 278          278 Second      Parkes, Mr. Francis "Frank" Male  NA     0
0
## 303          303  Third Johnson, Mr. William Cahoone Jr Male  19     0
0
## 414          414 Second   Cunningham, Mr. Alfred Fleming Male  NA     0
0
##     Ticket Fare Cabin    Embarked formula1 Family_Name
## 180   LINE    0       Southampton       Mr     Leonard
## 264 112059    0   B94 Southampton       Mr    Harrison
## 272   LINE    0       Southampton       Mr   Tornquist
## 278 239853    0       Southampton       Mr      Parkes
## 303   LINE    0       Southampton       Mr     Johnson
## 414 239853    0       Southampton       Mr  Cunningham
```

Todos son varones, mayores de edad y embarcaron en el puerto de Southampton. Cuando buscamos información de estos pasajeros vemos que algunos pertenecían al Titanic Guarantee Group (El equipo de Belfast enviado por los constructores de barcos Harland & Wolff para acompañar al Titanic en su viaje inaugural), con lo cual podemos suponer que los 0 son correctos, era gente que estaba viajando gratis.

Vemos que las variables `Age`, `Fare` y `Embarked` contienen valores perdidos.

Las ventajas de imputar son que logramos obtener un conjunto de datos completo sin datos faltantes, se puede reducir el sesgo debido a la no respuesta y la imputación opera sobre los datos, de forma que los resultados obtenidos por los diferentes análisis son mutuamente consistentes. Por otra parte, la imputación también tiene desventajas ya que hay que tener en cuenta que el futuro análisis no distingue entre las imputaciones y los datos reales. Además los valores imputados pueden ser buenas estimaciones pero no son datos reales y no podemos asegurar una mejora en el sesgo respecto del sistema de datos incompletos. Al fin y al cabo la imputación es un procedimiento para generar datos. Si el método de imputación no es el adecuado, posiblemente aumente el sesgo y sobreestime la varianza, obteniendo datos imputados inconsistentes produciendo una base de datos no confiables, llevando a la interpretación errónea de los resultados por parte de los usuarios.

Las variables `Fare` y `Embarked` tienen 1 y 2 valores perdidos respectivamente, como la muestras es bastante grande no hace falta imputar datos. Pero en la variable `Age` faltan 263 valores del 1309, representa un 20% de los datos, además un 20% es el máximo de valores perdidos para los que algunos autores recomiendan la imputación de datos.

**Realizaremos una imputación por Knn*

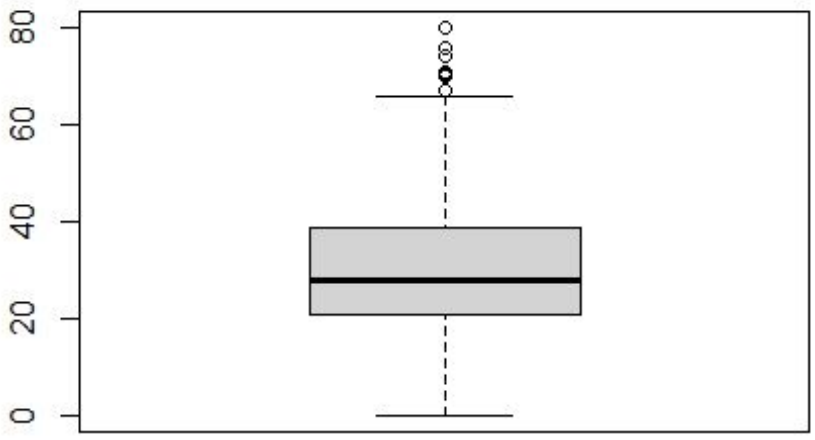
3.2. Identificación y tratamiento de valores extremos.

Los valores extremos o outliers son aquellos datos que se encuentran muy alejados de la distribución normal de una variable o población. Al ser observaciones que se desvían del resto levantan sospechas sobre si fueron generadas mediante el mismo mecanismo.

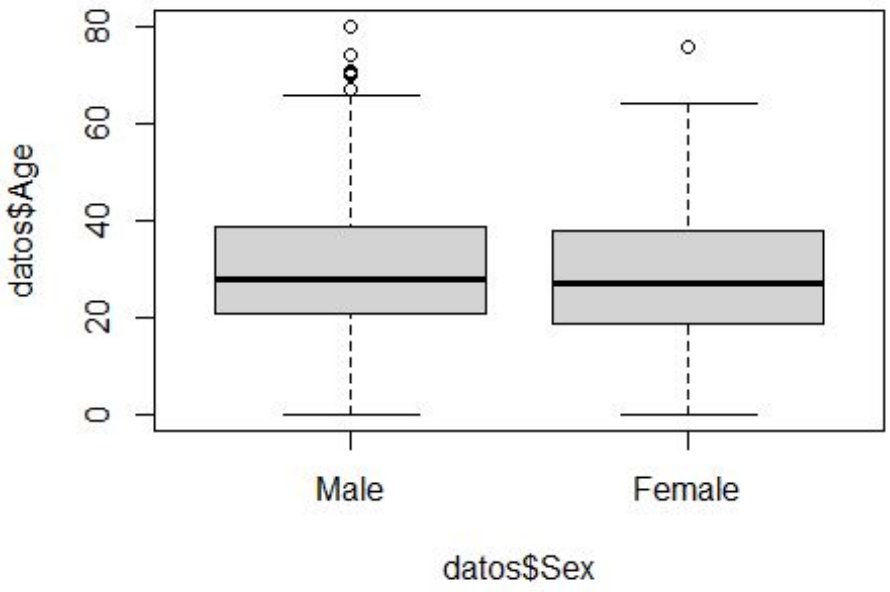
La decisión sobre qué se considera un valor extremo puede resultar controvertida, pero generalmente se considera que un valor es extremo cuando el valor se encuentra alejado 3 desviaciones estándar con respecto a la media del conjunto de datos. Por ello, normalmente se utiliza la representación de los datos mediante gráficos de cajas (boxplots) con el objetivo de detectar dichos outliers. Otros métodos que permiten detectar los valores extremos se basan en la distancia de Mahalanobis o la distancia de Cook.

Sus posibles efectos son: - incrementar el error en la varianza de los datos - sesgar los cálculos y estimaciones.

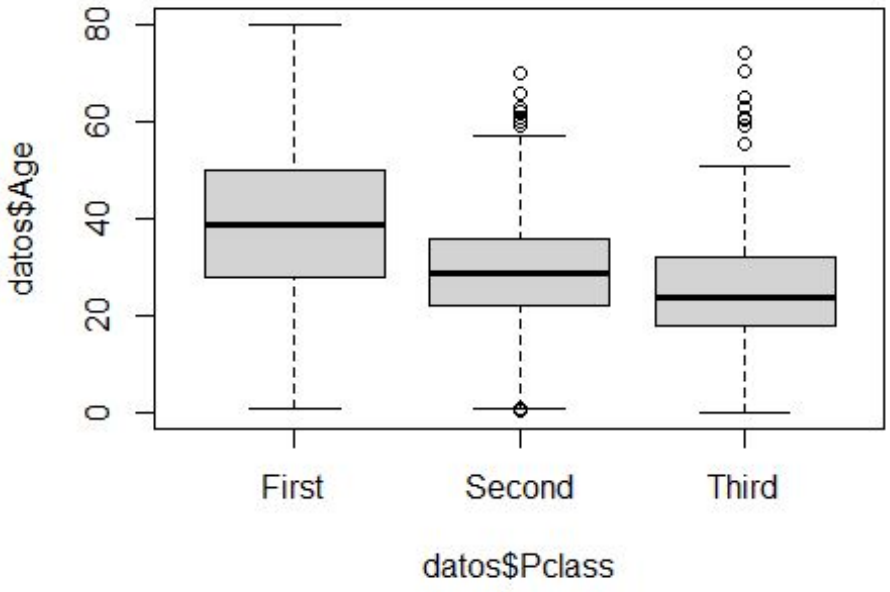
```
boxplot(datos$Age)
```



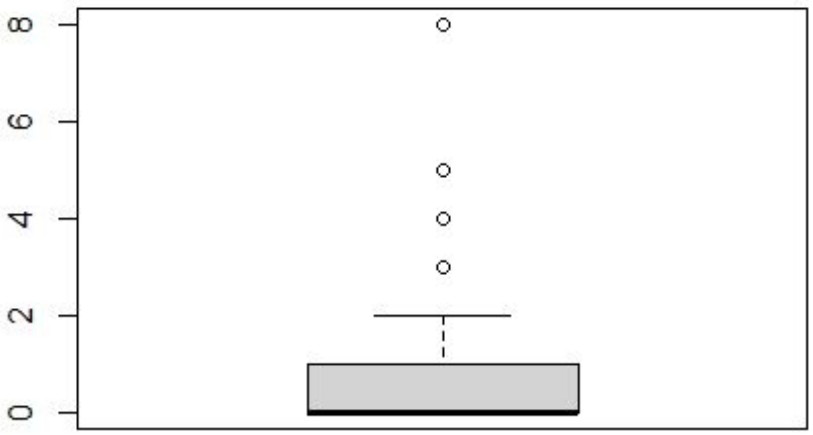

```
boxplot(datos$Age~datos$Sex)
```



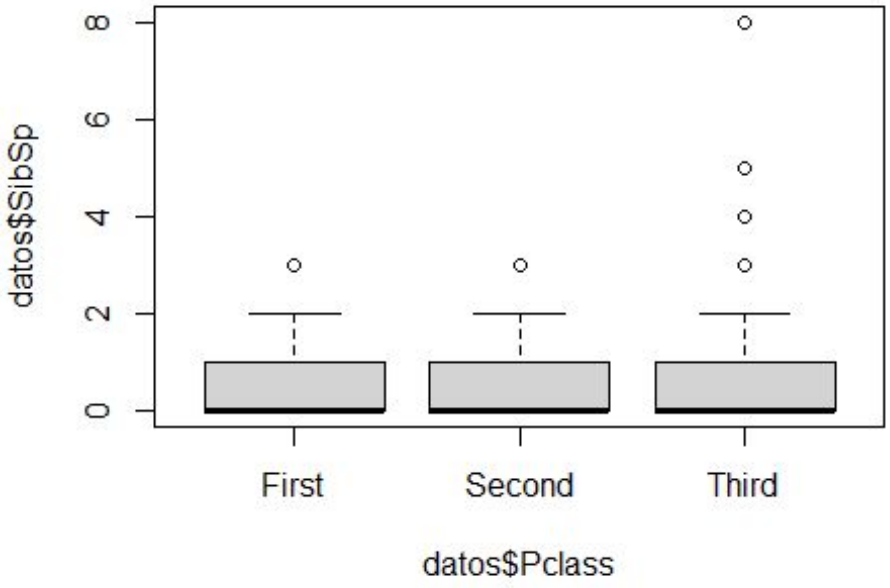

```
boxplot(datos$Age~datos$Pclass)
```



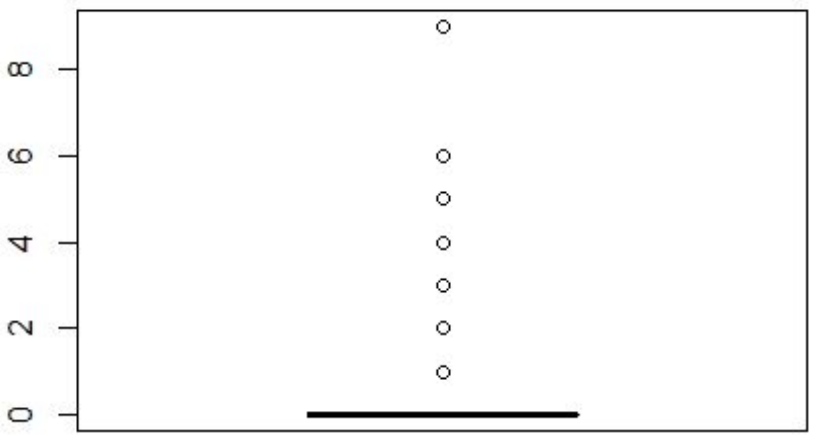

```
boxplot(datos$SibSp)
```



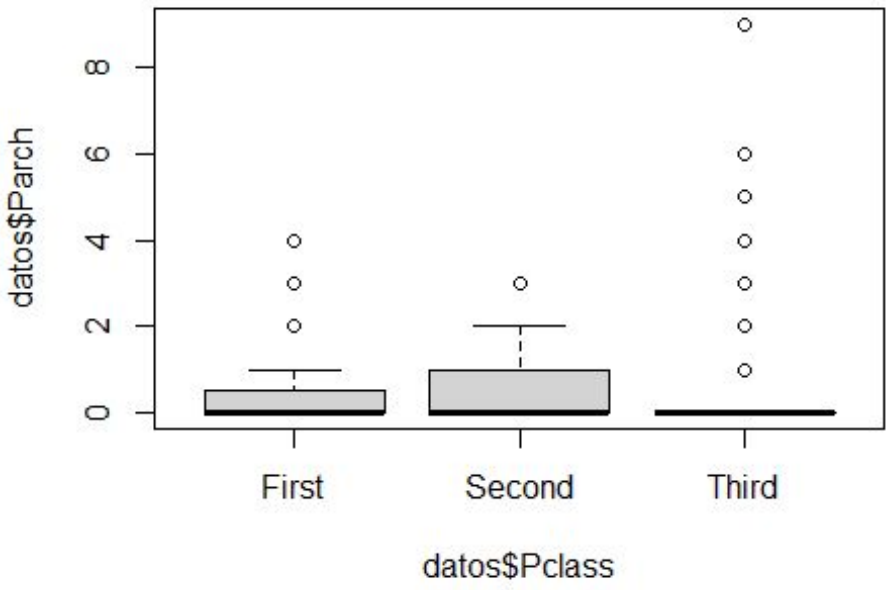

```
boxplot(datos$SibSp~datos$Pclass)
```



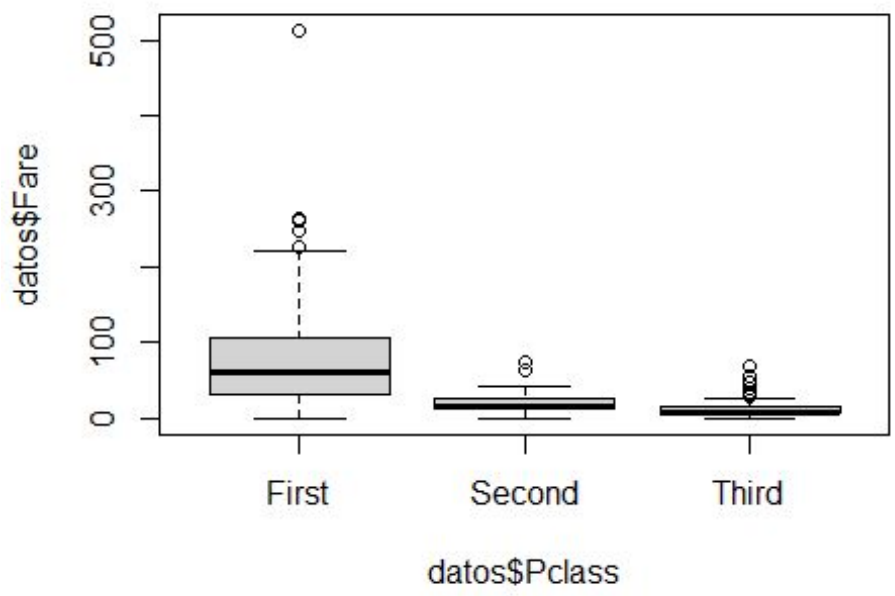

```
boxplot(datos$Parch)
```




```
boxplot(datos$Parch~datos$Pclass)
```




```
boxplot(datos$Fare~datos$Pclass)
```



Los valores extremos (outliers) pueden aparecer por distintos motivos. - son valores válidos que forman parte de la muestra - son valores debidos a una desviación sistemática en el grupo de valores extremos - son errores en los datos

En este caso parece que los outliers son valores válidos y entran dentro del rango de las variables, por tanto, forman parte de la muestra, por lo que no se deben modificar ni eliminar, y se deben tener en cuenta en el análisis de los datos.

4. Análisis de los datos.

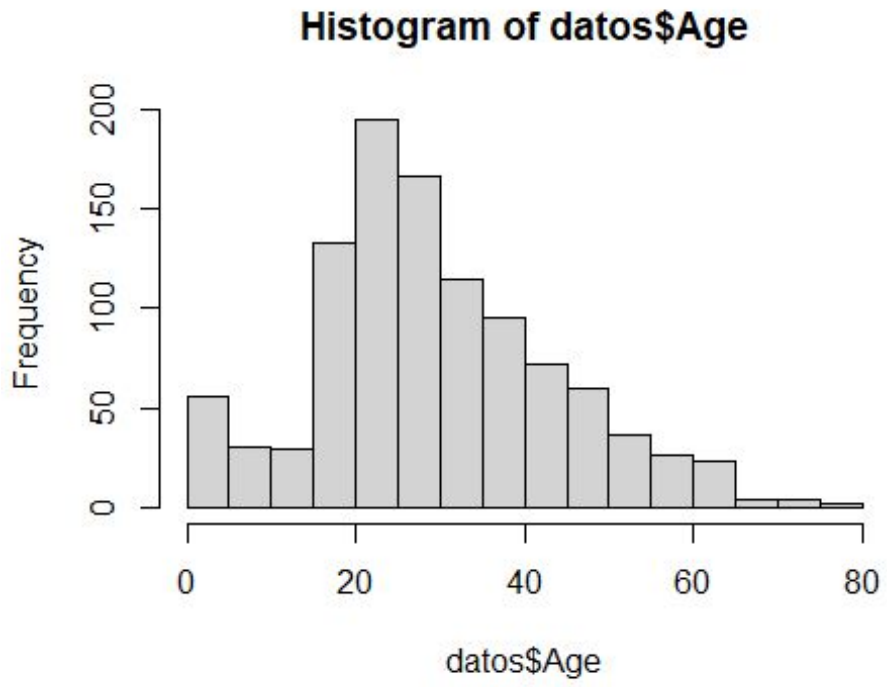
4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

- Predecir que pasajeros del dataset test sobrevivieron, usando el dataset train para entrenar un algoritmo de clasificación. En este caso parece interesante utilizar decision trees, ya que nos da la información de cómo se clasifican los pasajeros y nos devuelve un gráfico el mismo.
- Mediante un modelo de regresión lineal para predecir qué probabilidad hay de que un pasajero sobreviva en base a sus características.
- Diferencias entre la edad de los pasajeros de cada sexo (ttest).

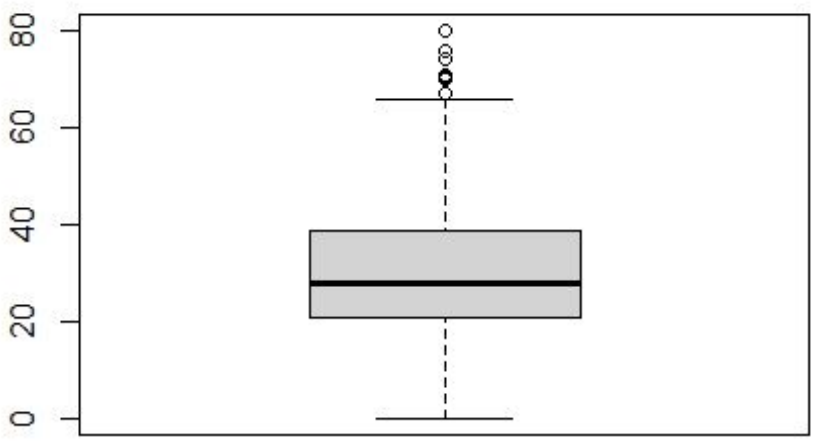
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

- Comprobación de la normalidad:

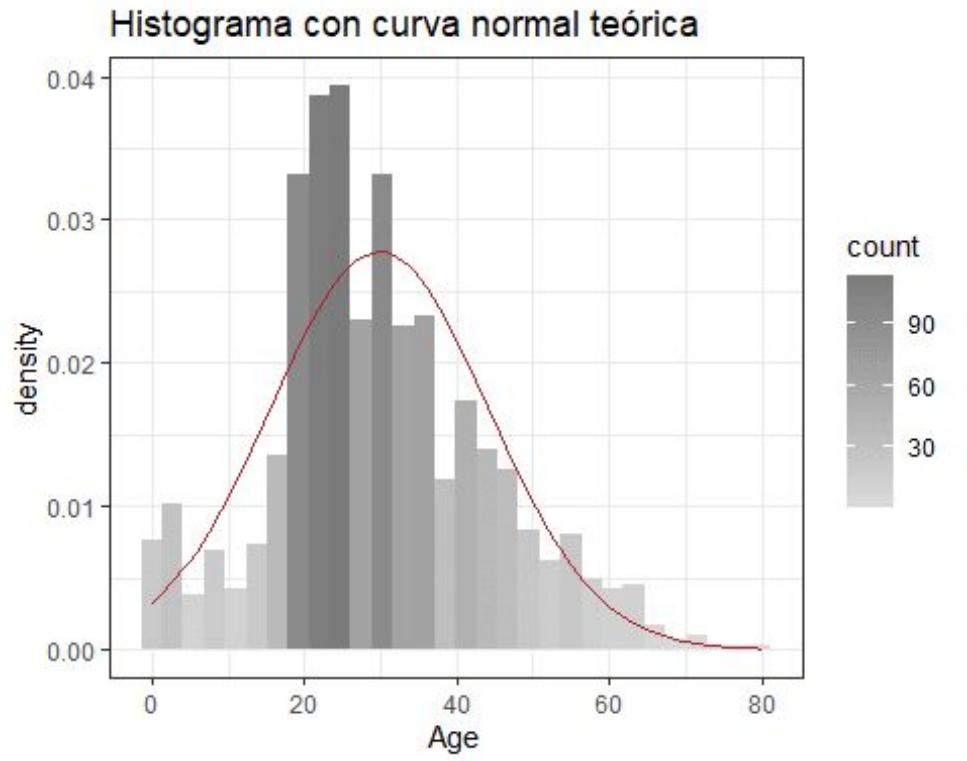
```
hist(datos$Age)
```



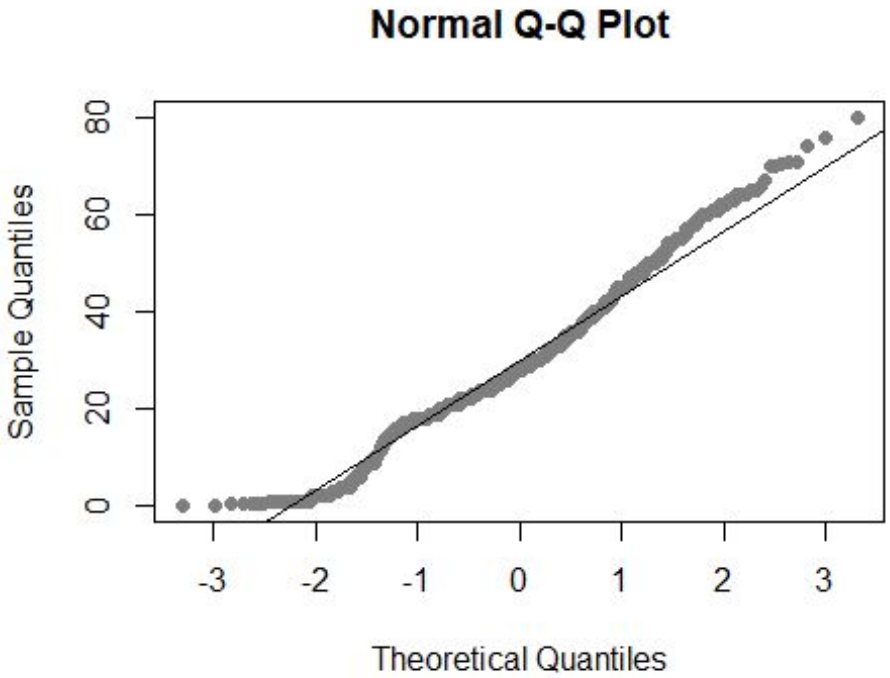

```r
boxplot(datos$Age)

library(ggplot2)
```



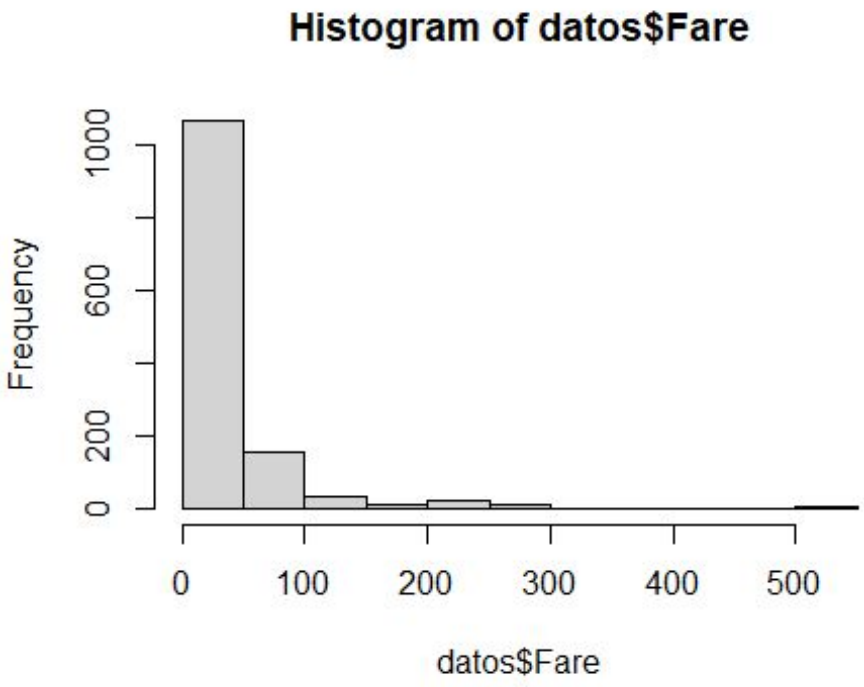

```
datos1 <- na.omit(datos)
ggplot(data = datos1, aes(x = Age)) +
geom_histogram(aes(y = ..density.., fill = ..count..)) +
scale_fill_gradient(low = "#DCDCDC", high = "#7C7C7C") +
stat_function(fun = dnorm, colour = "firebrick",args = list(mean =
mean(datos1$Age),sd = sd(datos1$Age))) +
ggtitle("Histograma con curva normal teórica") +
theme_bw()

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



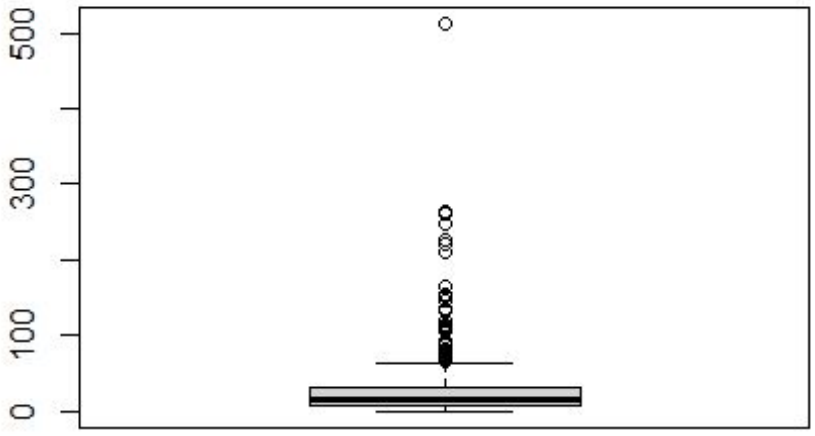

```r
qqnorm(datos$Age, pch = 19, col = "gray50")
qqline(datos$Age)
```



```r
hist(datos$Fare)
```

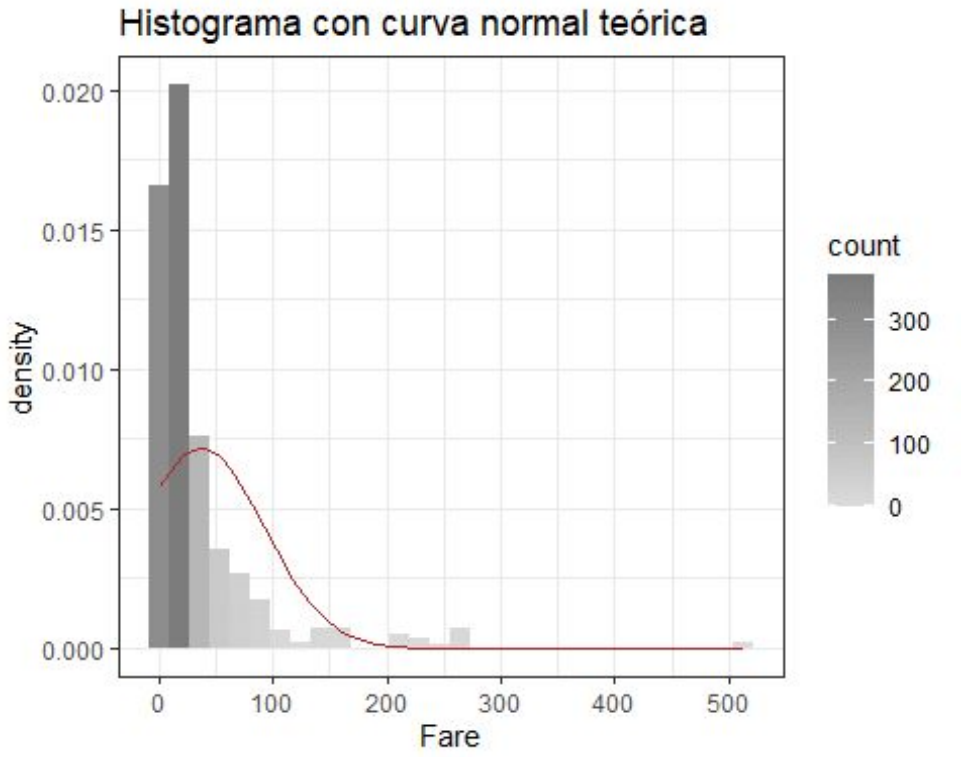



```r
boxplot(datos$Fare)
```

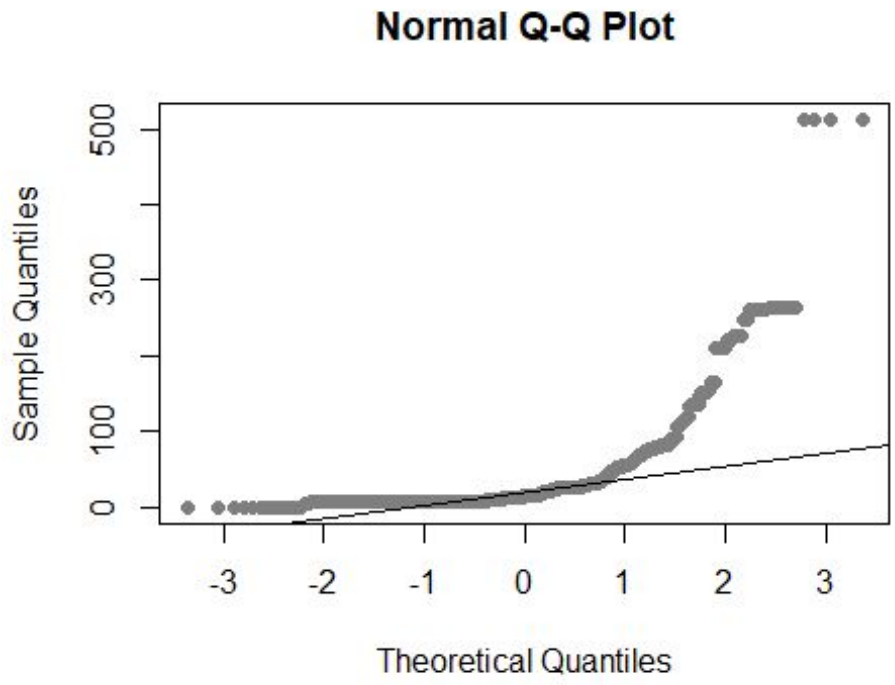


```r
library(ggplot2)
datos1 <- na.omit(datos)
ggplot(data = datos1, aes(x = Fare)) +
geom_histogram(aes(y = ..density.., fill = ..count..)) +
scale_fill_gradient(low = "#DCDCDC", high = "#7C7C7C") +
stat_function(fun = dnorm, colour = "firebrick",args = list(mean =
mean(datos1$Fare),sd = sd(datos1$Fare))) +
ggtitle("Histograma con curva normal teórica") +
theme_bw()
```

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




```r
qqnorm(datos$Fare, pch = 19, col = "gray50")
qqline(datos$Fare)
```



El test Lilliefors asume que la media y varianza son desconocidas, estando especialmente desarrollado para contrastar la normalidad. Es la alternativa al test de Shapiro-Wilk cuando el número de observaciones es mayor de 50, como es nuestro caso.

```
library("nortest")

## Warning: package 'nortest' was built under R version 4.0.3

lillie.test(x = datos$Age)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  datos$Age
## D = 0.078928, p-value < 2.2e-16

lillie.test(x = datos$Fare)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  datos$Fare
## D = 0.28586, p-value < 2.2e-16
```

- Comprobación homogeneidad de la varianza:

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

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

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