PrÃƒÂ¡ctica 2: Limpieza y validaciÃƒÂ³n de los datos

Fiol Bibiloni, Andreu

Navarro Yepes, JosÃƒÂ© AndrÃƒÂ©s

11 de junio de 2019

1. Descripción del *datatset*

El dataset escogido es llamado *Titanic*, obtenido de la página web de Kaggle (https://www.kaggle.com /c/titanic (https://www.kaggle.com/c/titanic)). Es un dataset recomendado por el enunciado de esta práctica, y sirve para la competición *Titanic: Machine Learning from Disaster*, organizada por la propia página y enmarcada en la categoría *Getting Started Prediction Competition*. Para su descarga hemos utilizado la API de Kaggle, que funciona con Python 3.

```
# Downloading the files of the competition. We need the Kaggle API (works with Pyt
hon 3)
system("kaggle competitions download -c titanic")

# Loading the training file
trainData <- read.csv("train.csv")

# Loading the test file
testData <- read.csv("test.csv")
```

Es un dataset muy utilizado porque marca una de las tragedias internacionales más conocidas de la historia (en parte gracias a James Cameron) y porque para no deja de ser un dataset útil para practicar aprendizaje automático mediante algún lenguaje de programación como Python o R. La pregunta más importante a la que intentamos responder es a predecir cuántos pasajeros y qué tipo de pasajero sobreviviría a esta catástrofe. Derivada de ésta podremos contestar a otras preguntas más curiosas como la clase y el sexo de los supervivientes

Es un dataset que está ya dividido en dos partes: una de entrenamiento y otra de prueba. Veamos las características del grupo de entrenamiento.

```
str(trainData)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : 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 2
76 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 ...
```

```
summary(trainData)
```

```
##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##                                     Name         Sex           Age
##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00
##  Abelson, Mr. Samuel                   :  1                Mean   :29.70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00
##  Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.00
##  (Other)                               :885                NA's   :177
##      SibSp           Parch             Ticket         Fare
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33
##                                   (Other) :852
##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
##  (Other)    :186
```

Veamos ahora las características del grupo de prueba:

```
str(testData)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 40
9 273 414 182 370 85 58 5 104 ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ 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     : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 2
62 159 85 101 270 ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```
summary(testData)
```

```
##   PassengerId         Pclass
##  Min.   : 892.0   Min.   :1.000
##  1st Qu.: 996.2   1st Qu.:1.000
##  Median :1100.5   Median :3.000
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##                                          Name         Sex
##  Abbott, Master. Eugene Joseph            :  1   female:152
##  Abelseth, Miss. Karen Marie              :  1   male  :266
##  Abelseth, Mr. Olaus Jorgensen            :  1
##  Abrahamsson, Mr. Abraham August Johannes :  1
##  Abrahim, Mrs. Joseph (Sophie Halaut Easu):  1
##  Aks, Master. Philip Frank                :  1
##  (Other)                                  :412
##       Age            SibSp            Parch             Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   PC 17608:  5
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   113503  :  4
##  Median :27.00   Median :0.0000   Median :0.0000   CA. 2343:  4
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923   16966   :  3
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000   220845  :  3
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000   347077  :  3
##  NA's   :86                                        (Other) :396
##       Fare                     Cabin     Embarked
##  Min.   :  0.000                 :327   C:102
##  1st Qu.:  7.896   B57 B59 B63 B66:  3   Q: 46
##  Median : 14.454   A34           :  2   S:270
##  Mean   : 35.627   B45           :  2
##  3rd Qu.: 31.500   C101          :  2
##  Max.   :512.329   C116          :  2
##  NA's   :1         (Other)       : 80
```

Como podemos comprobar, el conjunto de prueba carece de la variable objetivo *Survived*. Por lo tanto, nuestro objetivo será predecir correctamente si los pasajeros del grupo de prueba sobrevivieron o no, a partir de sus datos.

En este proyecto se han utilizado ambos subconjuntos. Por una parte, con el training set, como su nombre indica, se pretende entrenar los datos para en la competición crear un modelo de Machine Learning. Por otra parte, el test set es más bien para comprobar cómo se desempeñan los datos con 'unseen data'. Acerca de ls variables mencionaremos que la columna Survived representa si el pasajero sobrevivió (1) o no (0); la columna de edad representa los años dl pasajero, passenger class (pclass) representa la clase en la que viajaban (primera clase = 1, segunda = 2 y tercera = 3); SibSp es el número de hermanos/cónyuges a bordo del Titanic; Parch es número de padres/niños a bordo (si un niño tiene 0 significa que viajaban sin padres pero con niñera); ticket representa el número de ticket; fare es la tarifa del pasajero (precio del pasaje); cabin es el número de cabina de cada pasajero; embarked es el puerto de embarque, pudiendo ser tres tipos de puertos C = Cherbourg, Q = Queenstown y S = Southampton y el passengerid es el identificador único atribuido a cada pasajero en forma de número entero, siendo la clave primaria de la tabla.

En cuanto a SibSp y Parch es conveniente clarificar los roles que se han tenido en cuenta en los datos:

● Hermano: hermano, hermana, hermanastro o hermanastra del pasajero a bordo del Titanic ● Cónyuge: esposo o esposa del pasajero a bordo del Titanic (amantes y novios se han ignorado o se desconoce) ● Padre: Madre o padre del pasajero a bordo del Titanic ● Niño: hijo, hija, hijastro o hijastra del pasajero a bordo del Titanic

2. Integración y selección de los datos de interés

En primer lugar uniremos los dos subconjuntos para disponer de un dataset completo.

```
# Unión de los subconjuntos de datos
data <- merge(trainData, testData, all = T)
str(data)
```

```
## 'data.frame':    1309 obs. of  12 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       : Factor w/ 1307 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/ 929 levels "110152","110413",..: 524 597 670 50 473 2
76 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 187 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 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
```

```
summary(data)
```

```
##   PassengerId       Pclass                                      Name
##  Min.   :   1   Min.   :1.000   Connolly, Miss. Kate            :   2
##  1st Qu.: 328   1st Qu.:2.000   Kelly, Mr. James                :   2
##  Median : 655   Median :3.000   Abbing, Mr. Anthony             :   1
##  Mean   : 655   Mean   :2.295   Abbott, Mr. Rossmore Edward     :   1
##  3rd Qu.: 982   3rd Qu.:3.000   Abbott, Mrs. Stanton (Rosa Hunt):   1
##  Max.   :1309   Max.   :3.000   Abelson, Mr. Samuel             :   1
##                                 (Other)                         :1301
##      Sex           Age            SibSp            Parch
##  female:466   Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  male  :843   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##               Median :28.00   Median :0.0000   Median :0.000
##               Mean   :29.88   Mean   :0.4989   Mean   :0.385
##               3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##               Max.   :80.00   Max.   :8.0000   Max.   :9.000
##               NA's   :263
##       Ticket          Fare                     Cabin      Embarked
##  CA. 2343:  11   Min.   :  0.000                  :1014    :  2
##  1601    :   8   1st Qu.:  7.896   C23 C25 C27    :   6   C:270
##  CA 2144 :   8   Median : 14.454   B57 B59 B63 B66:   5   Q:123
##  3101295 :   7   Mean   : 33.295   G6             :   5   S:914
##  347077  :   7   3rd Qu.: 31.275   B96 B98        :   4
##  347082  :   7   Max.   :512.329   C22 C26        :   4
##  (Other) :1261   NA's   :1         (Other)        : 271
##     Survived
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3838
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :418
```

Pasamos ahora a seleccionar los datos que nos interesan de cada pasajero. Entre ellos se encuentran el sexo, la edad, la clase, el número de familiares, el precio del pasaje, el lugar de embrque y si sobrevivió o no. No nos interesarán sus nombres, número de ticket, número de pasajero ni número de cabina.

```
# Selección de las variables que nos interesan
data <- data[, c(-1, -3, -8, -10)]

# Comprobación
str(data)
```

```
## 'data.frame':    1309 obs. of  8 variables:
##  $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp   : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
```

```
summary(data)
```

```
##      Pclass          Sex           Age            SibSp
##  Min.   :1.000   female:466   Min.   : 0.17   Min.   :0.0000
##  1st Qu.:2.000   male  :843   1st Qu.:21.00   1st Qu.:0.0000
##  Median :3.000                Median :28.00   Median :0.0000
##  Mean   :2.295                Mean   :29.88   Mean   :0.4989
##  3rd Qu.:3.000                3rd Qu.:39.00   3rd Qu.:1.0000
##  Max.   :3.000                Max.   :80.00   Max.   :8.0000
##                               NA's   :263
##      Parch            Fare          Embarked    Survived
##  Min.   :0.000   Min.   :  0.000    :  2    Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:  7.896   C:270    1st Qu.:0.0000
##  Median :0.000   Median : 14.454   Q:123    Median :0.0000
##  Mean   :0.385   Mean   : 33.295   S:914    Mean   :0.3838
##  3rd Qu.:0.000   3rd Qu.: 31.275            3rd Qu.:1.0000
##  Max.   :9.000   Max.   :512.329            Max.   :1.0000
##                  NA's   :1                  NA's   :418
```

3. *Data cleaning*

Pasamos ahora a la limpieza de los datos.

3.1. Elementos vacíos

Por supuesto, nos encontramos con un buen número de datos vacíos en la columna *Survived* debido a que hemos añadido el suconjunto de prueba sin ese dato. Deberemos hacer frente a ello y a otros posibles datos vacíos. para ello utilizaremos el método missForest, por ser así recomendado por Calvo, Subirats y Pérez (2019).

```
# Comprueba las variables con valores perdidos
data[data==""] <- NA
names(which(sapply(data, anyNA)))
```

```
## [1] "Age"      "Fare"     "Embarked" "Survived"
```

```
# Imputación
mydata <- missForest(data, variablewise = T)
```

```
##   missForest iteration 1 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you want
## to do regression?
```

```
## done!
##   missForest iteration 2 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you want
## to do regression?
```

```
## done!
##   missForest iteration 3 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you want
## to do regression?
```

```
## done!
```

```
# Comprueba que ya no hay valores perdidos
which(is.na(mydata))
```

```
## named integer(0)
```

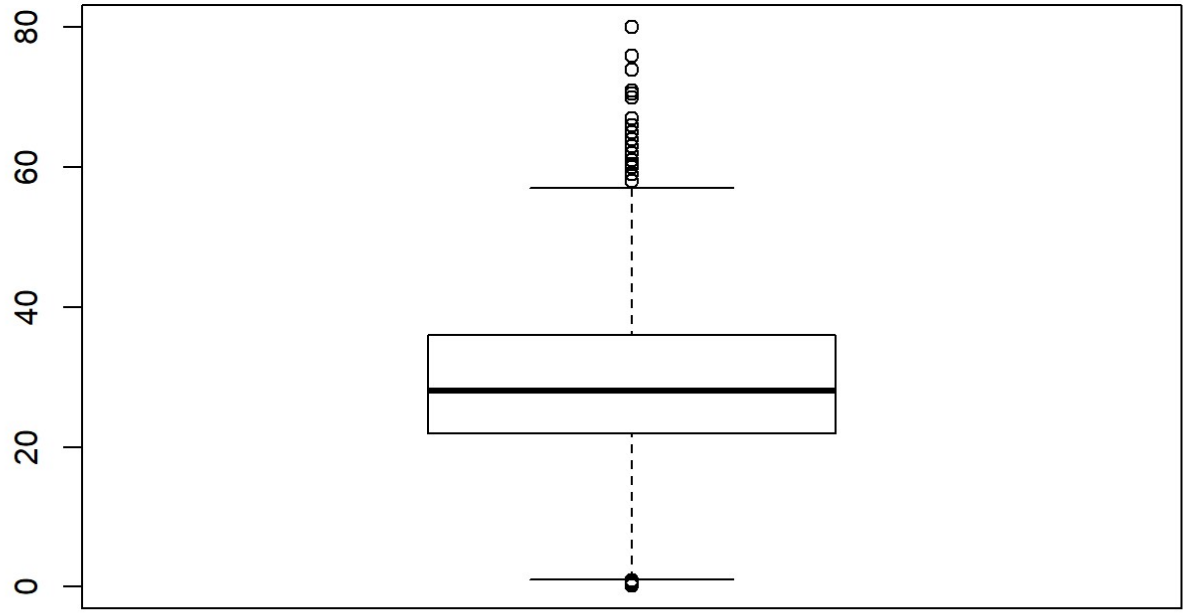
```
# Nuevo data frame sobre el que trabajaremos
datai <- mydata$ximp
attach(datai)
```

3.2. *Outliers*

Una vez tratados los datos vacíos observamos los posibles outliers.

Veamos el boxplot para la edad:

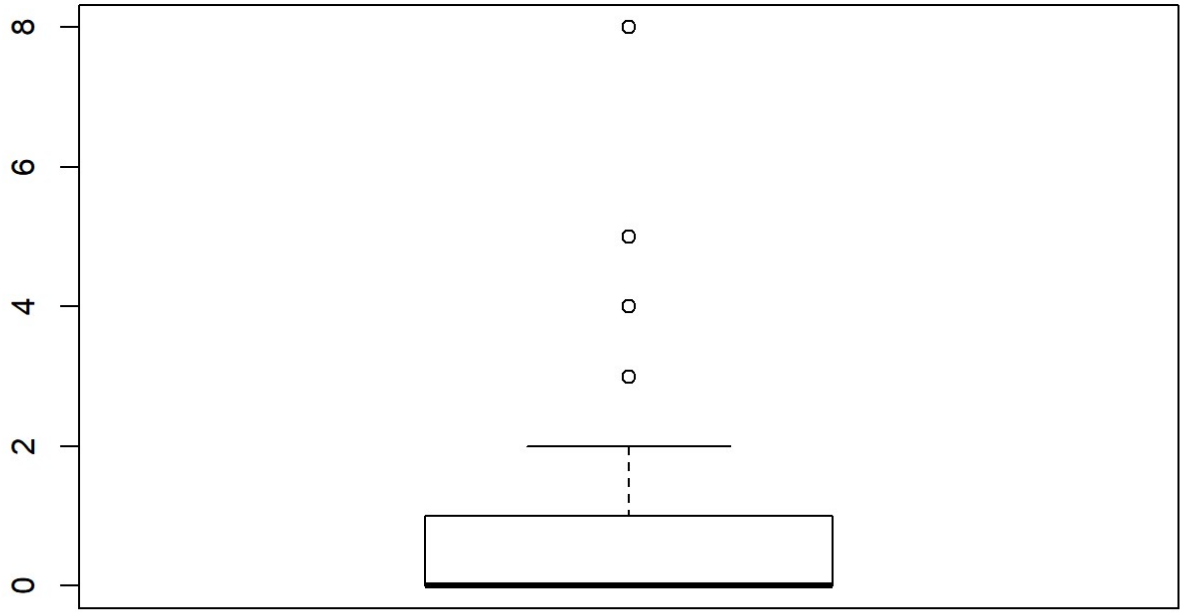
```
boxplot(Age)
```



Observamos que, aunque se aprecian diversos outliers, son siempre edades que se encuentran dentro de lo razonable, por lo que consideramos que estos datos no precisan de más tratamiento.

Pasemos a los daos de hermanos y pareja:

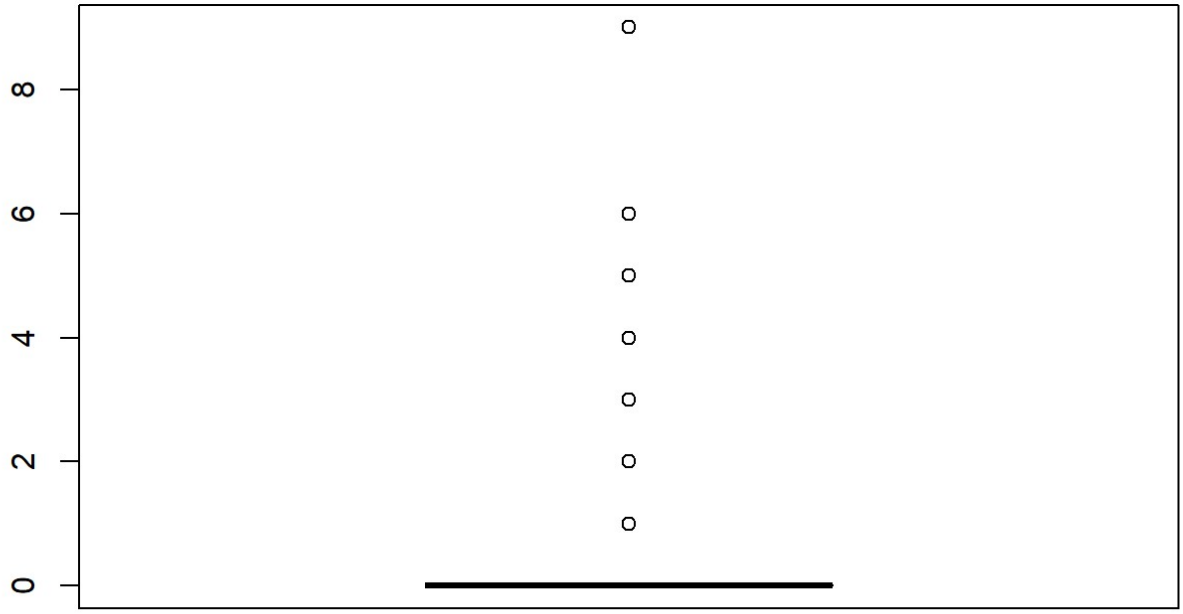
```
boxplot(SibSp)
```



De nuevo, aunque tenemos ciertos valores extremos, niguno se sale de lo que es razonablemente admisible, por lo que dejaremos los valores.

Veamos también el boxplot para el número de padres e hijos:

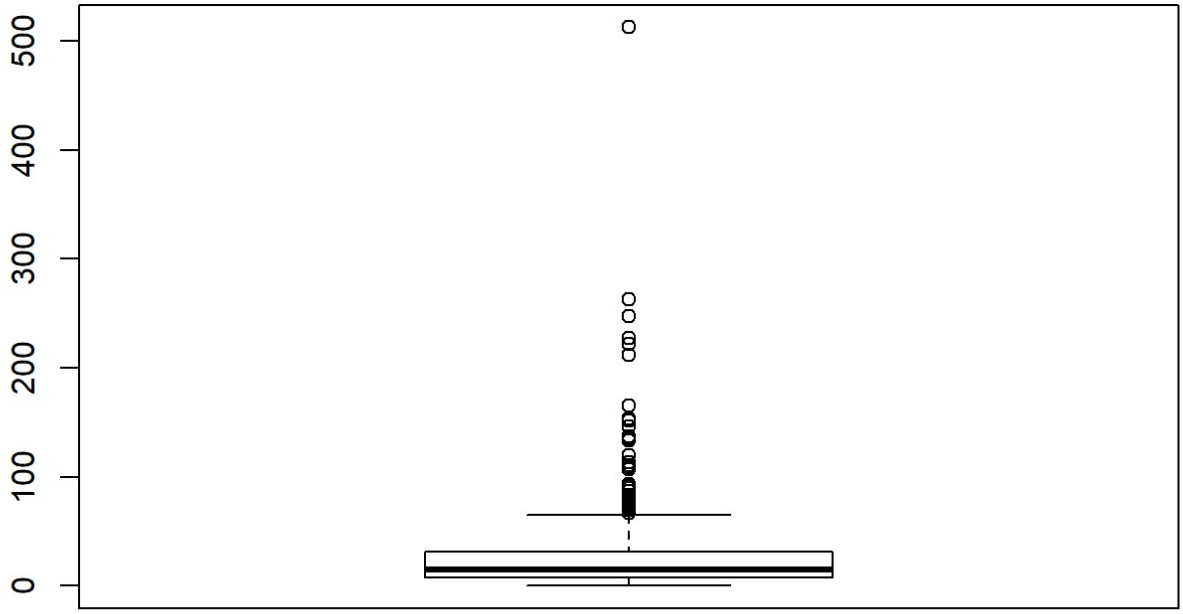
```
boxplot(Parch)
```



Una vez más nos encontramos con valores asumibles.

POr último observaremos el boxplot para el precio del billete:

```
boxplot(Fare)
```



A pesar del gran número de valores extremos, se trata de una caraterística de los precios en presencia de bienes de lujo como lo fue este viaje inaugural del Titanic.

Concluimos, pues, que en nuestro dataset no hemos hallado la necesidad de actuar sobre los valores extremos.

4. *Data analysis*

Disponemos finalmente de un conjunto de datos *limpio* sobre el que realizar análisis de datos. Seguidamente podemos observar las características más importantes de este conjunto:

```
# Análisis descriptivo
summary(datai)
```

```
##      Pclass          Sex           Age            SibSp
##  Min.   :1.000   female:466   Min.   : 0.17   Min.   :0.0000
##  1st Qu.:2.000   male  :843   1st Qu.:22.00   1st Qu.:0.0000
##  Median :3.000                Median :28.08   Median :0.0000
##  Mean   :2.295                Mean   :29.69   Mean   :0.4989
##  3rd Qu.:3.000                3rd Qu.:36.00   3rd Qu.:1.0000
##  Max.   :3.000                Max.   :80.00   Max.   :8.0000
##      Parch            Fare          Embarked    Survived
##  Min.   :0.000   Min.   :  0.000    :  0    Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:  7.896   C:272    1st Qu.:0.0000
##  Median :0.000   Median : 14.454   Q:123    Median :0.1441
##  Mean   :0.385   Mean   : 33.279   S:914    Mean   :0.3933
##  3rd Qu.:0.000   3rd Qu.: 31.275            3rd Qu.:1.0000
##  Max.   :9.000   Max.   :512.329            Max.   :1.0000
```

```
str(datai)
```

```
## 'data.frame':    1309 obs. of  8 variables:
##  $ Pclass  : num  3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 ...
##  $ SibSp   : num  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : num  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Survived: num  0 1 1 1 0 0 0 0 1 1 ...
```

4.1. Planifación de los análisis

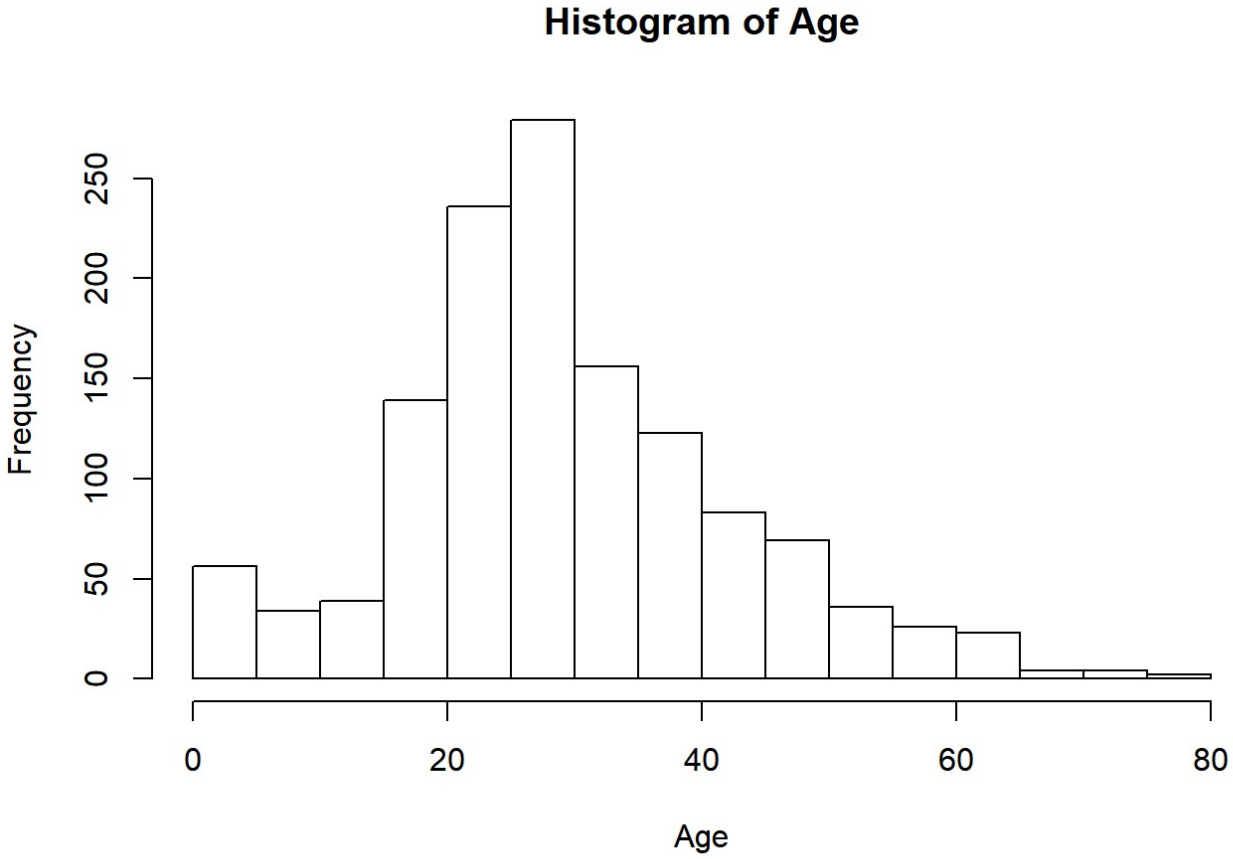
Procederemos a realizar tres análsis que responden a tres preguntas que este dataset podría resolver. 1: ¿Existe discriminación de precios por razón del sexo en el Titanic? Utilizaremos un contraste de hipótesis. 2: ¿Qué modelo rige el precio de los pasajes? Utilizaremos una regresión lineal múltiple. 3: ¿Podremos predecir qué pasajeros sobrevivieron a partir de estos datos? Para ello utilizaremos una regresión logística.

4.2. Normalidad

Antes de proceder a las pruebas estadísticas debemos conocer si nuestras variables numéricas cumplen la condición de normalidad.

Empecemos con la edad:

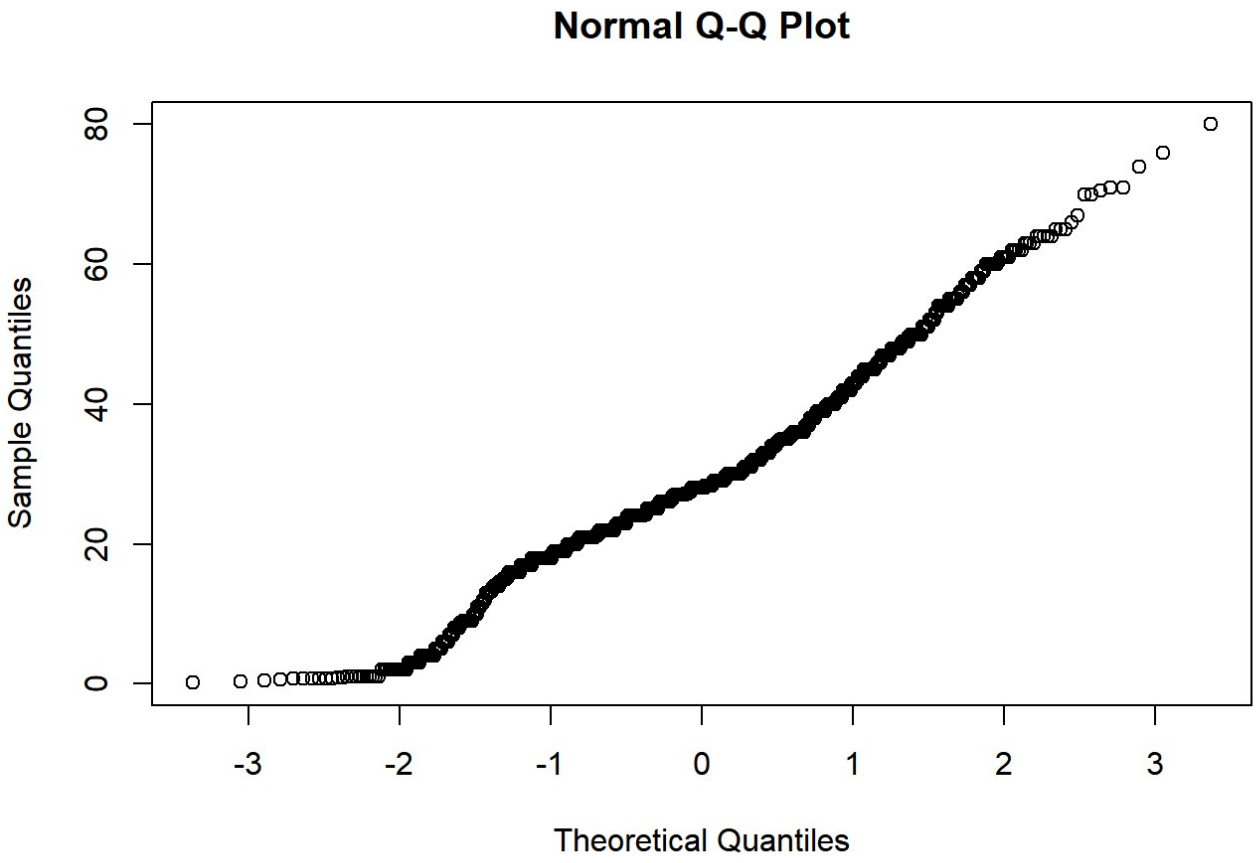
```
# Histograma
hist(Age)
```



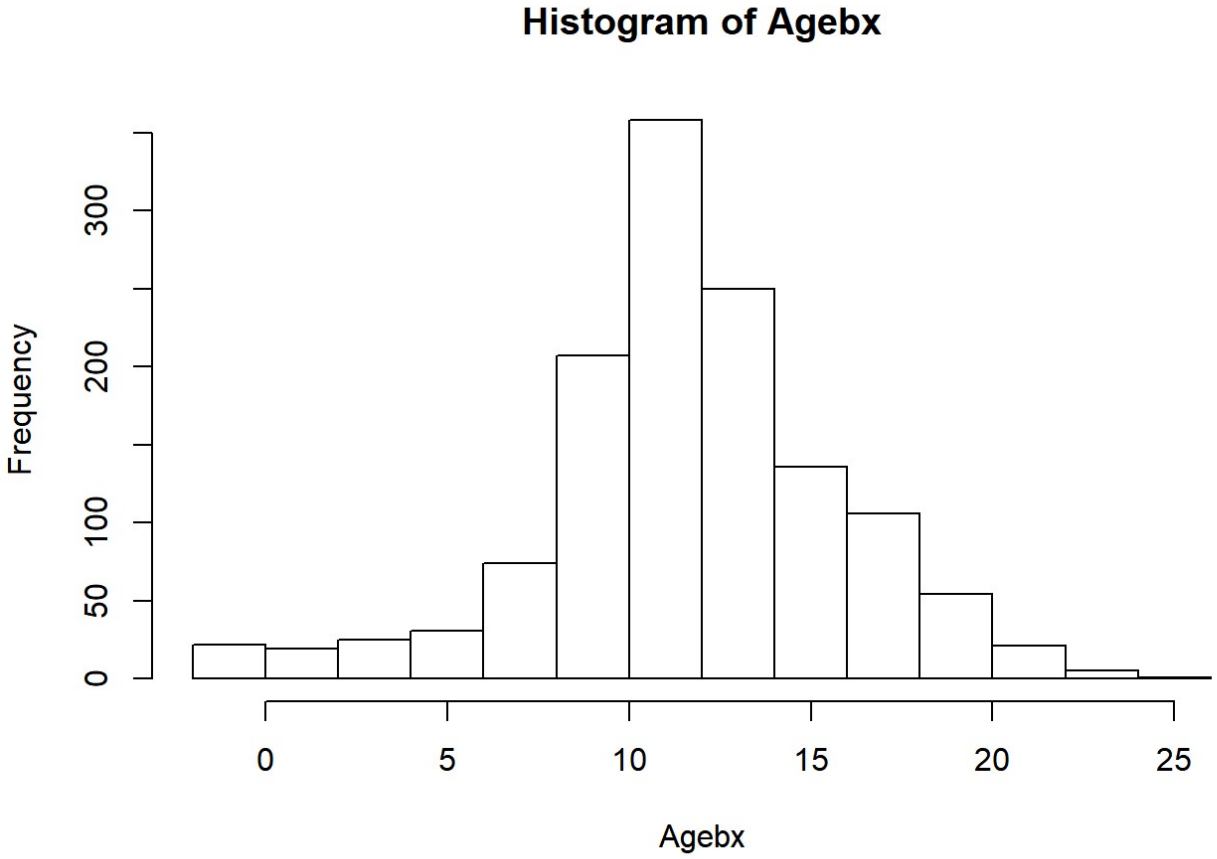

```
# Test de Shapiro
shapiro.test(Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Age
## W = 0.97623, p-value = 7.218e-14
```

```
# Gráfica Q-Q
qqnorm(Age)
```



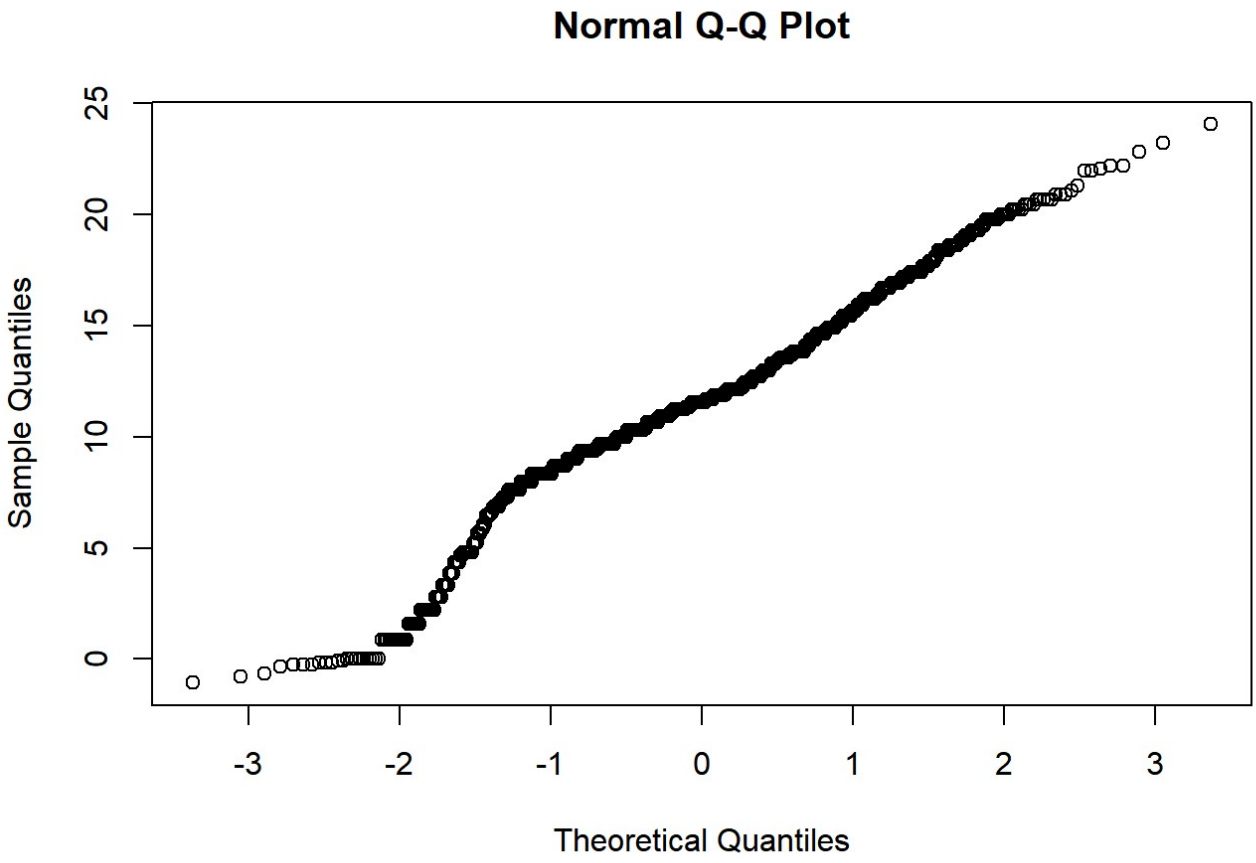

```
# Box-Cox
Agebx <- BoxCox(Age, lambda = BoxCoxLambda(Age))
hist(Agebx)
```




```
shapiro.test(Agebx)
```

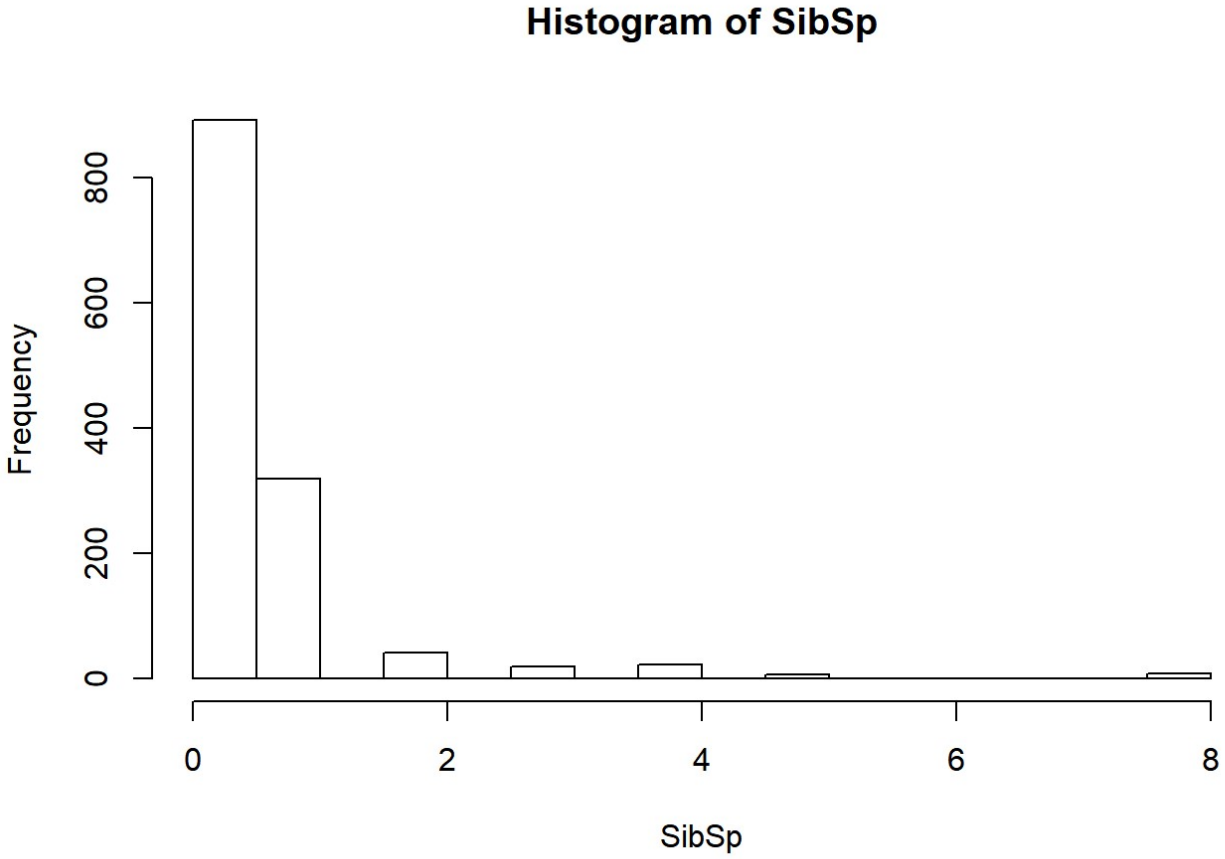
```
##
##  Shapiro-Wilk normality test
##
## data:  Agebx
## W = 0.97388, p-value = 1.193e-14
```

```
qqnorm(Agebx)
```



Sigamos con los hermanos y cónyuges:

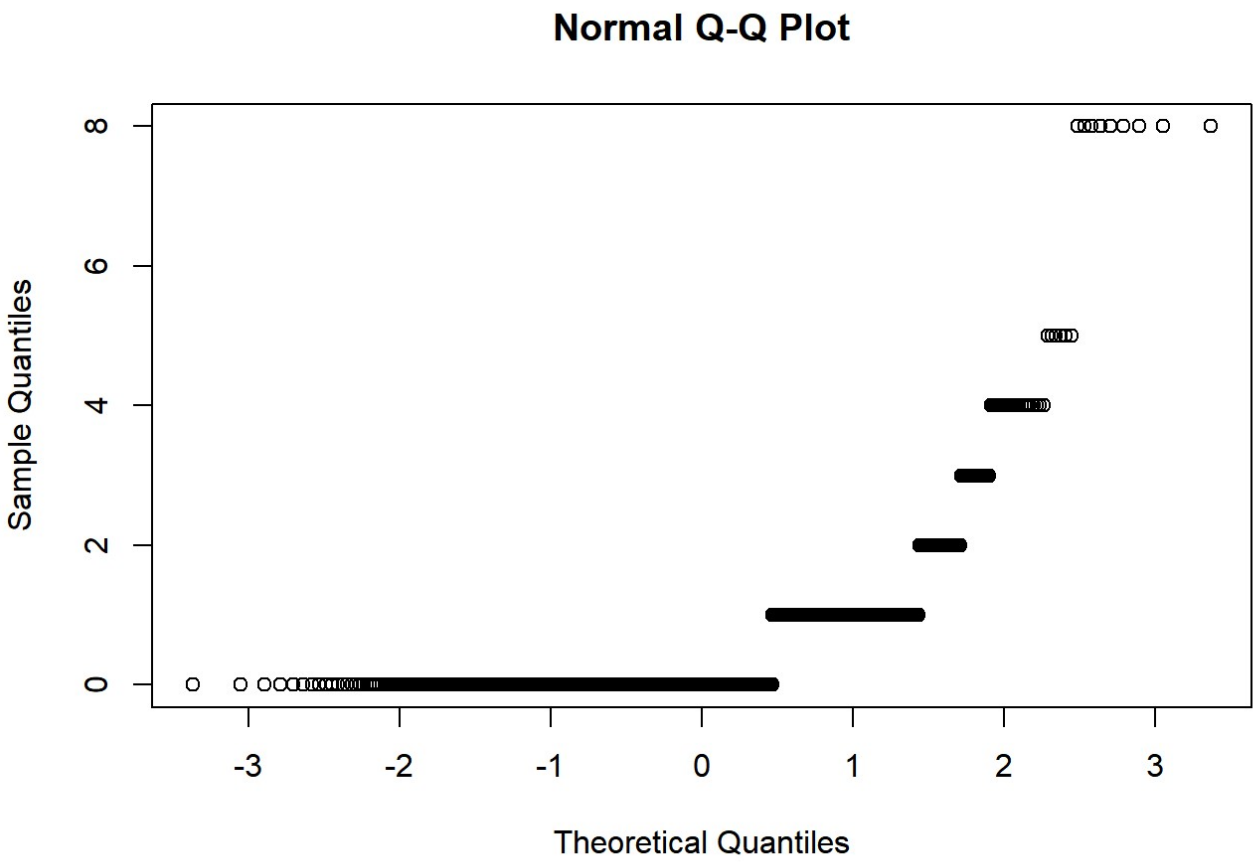
```
hist(SibSp)
```



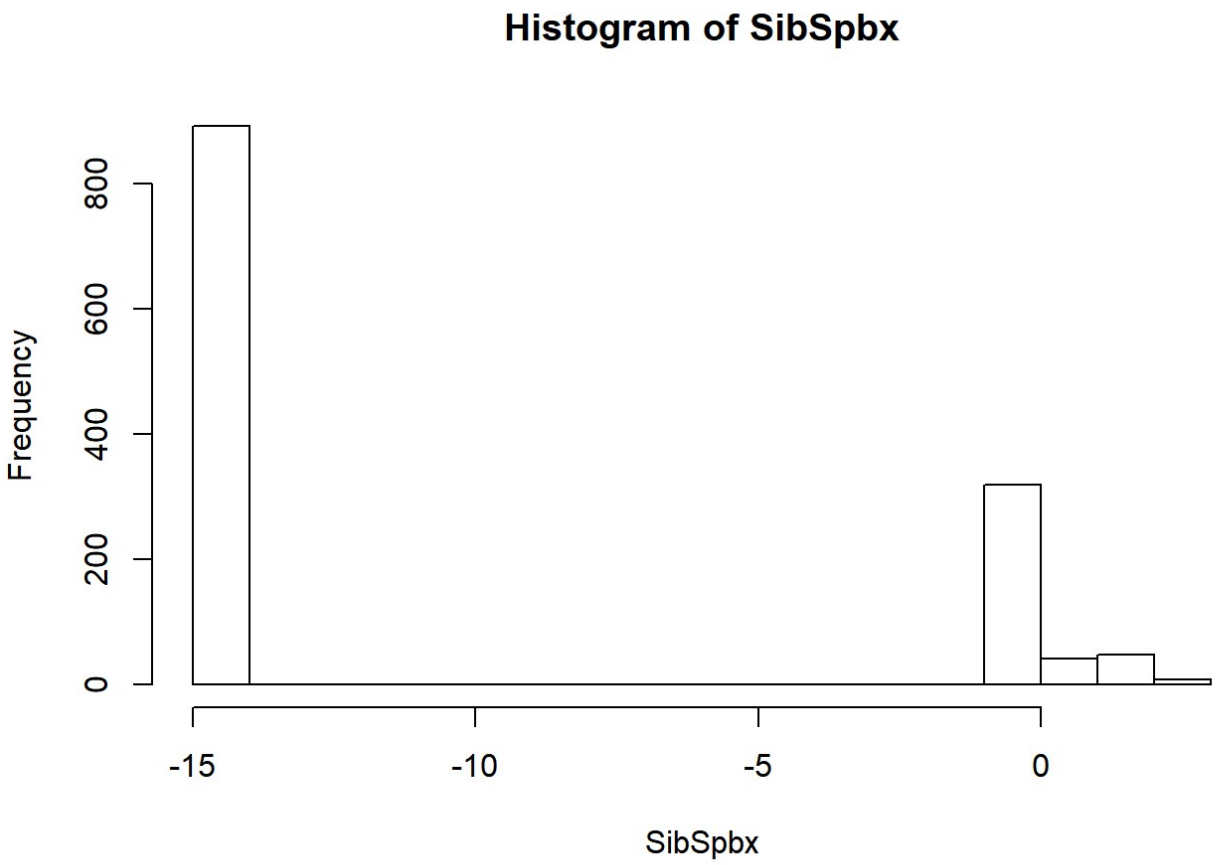

```
shapiro.test(SibSp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  SibSp
## W = 0.51108, p-value < 2.2e-16
```

```
qqnorm(SibSp)
```



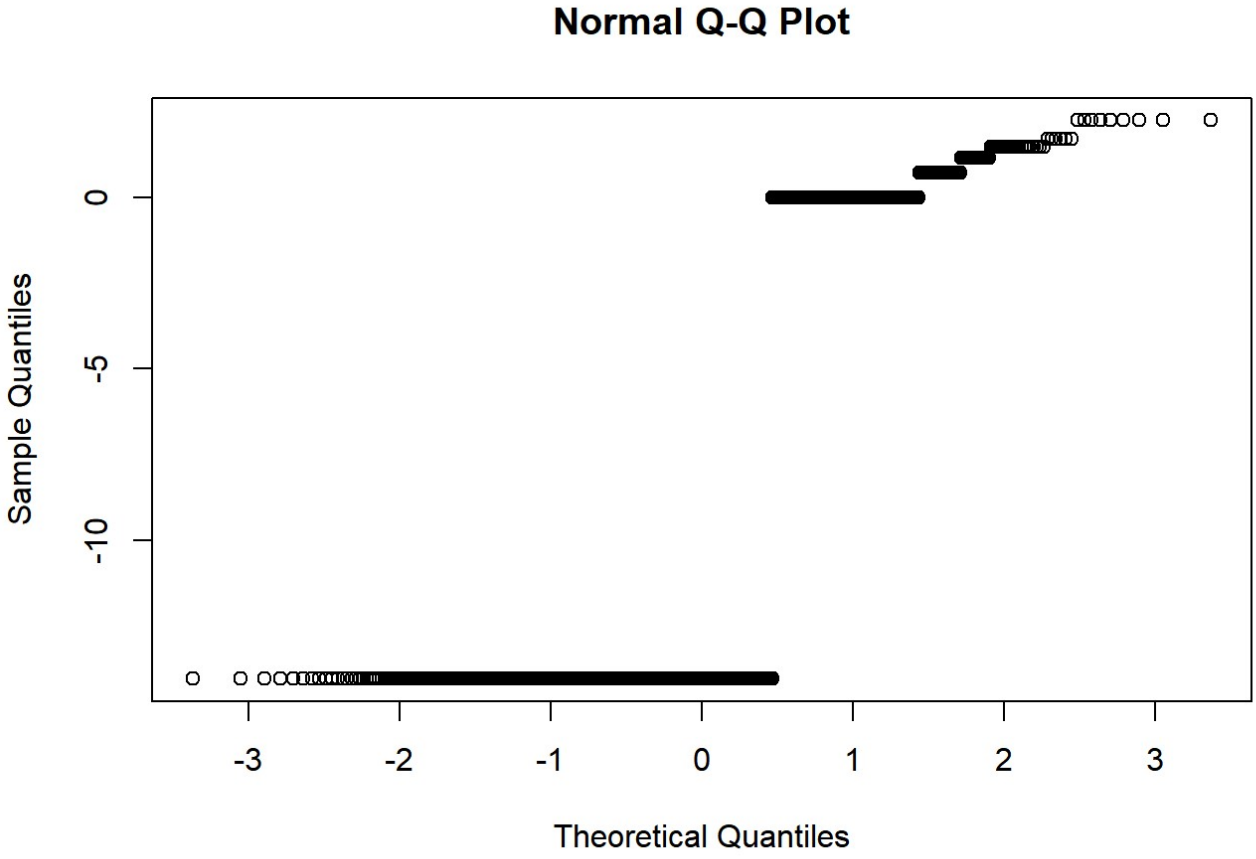

```
SibSpbx <- BoxCox(SibSp, lambda = BoxCoxLambda(SibSp))
hist(SibSpbx)
```




```
shapiro.test(SibSpbx)
```

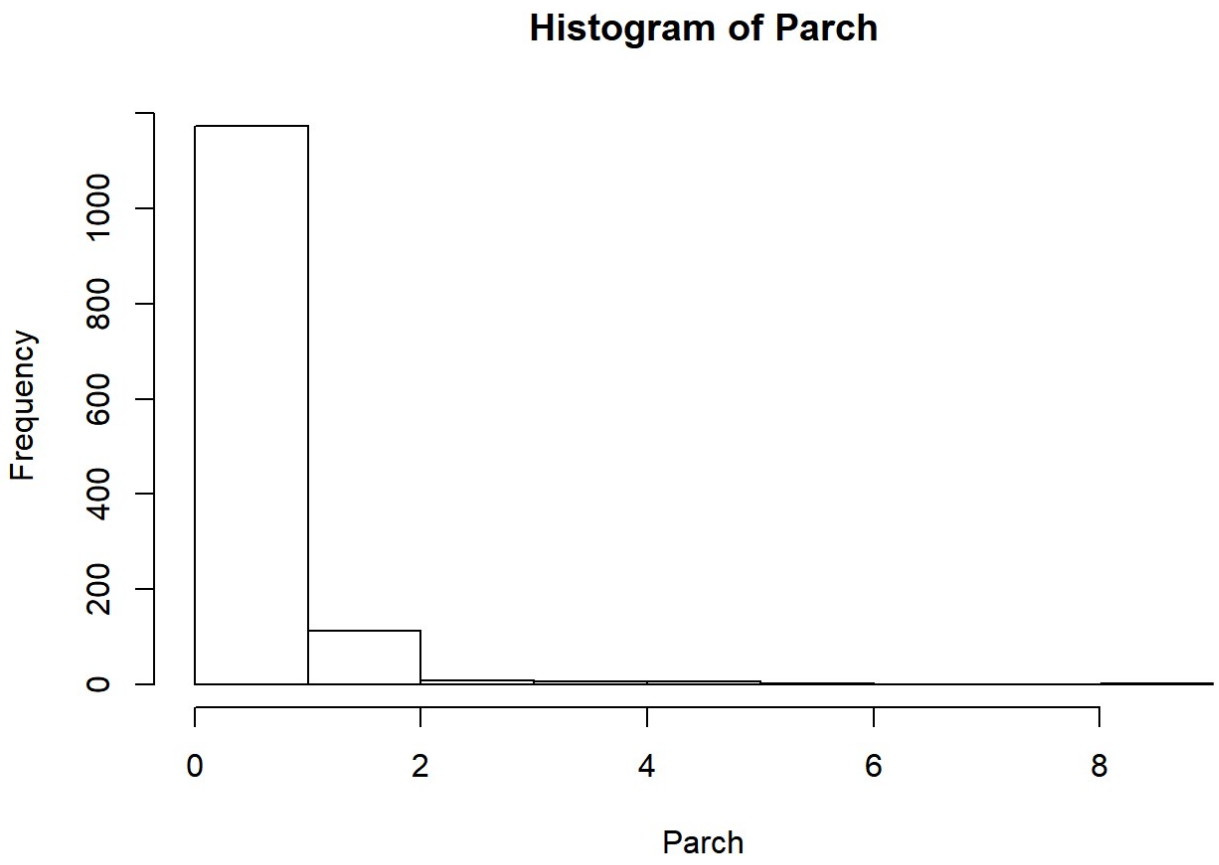
```
##
##  Shapiro-Wilk normality test
##
## data:  SibSpbx
## W = 0.60469, p-value < 2.2e-16
```

```
qqnorm(SibSpbx)
```



Hijos y padres:

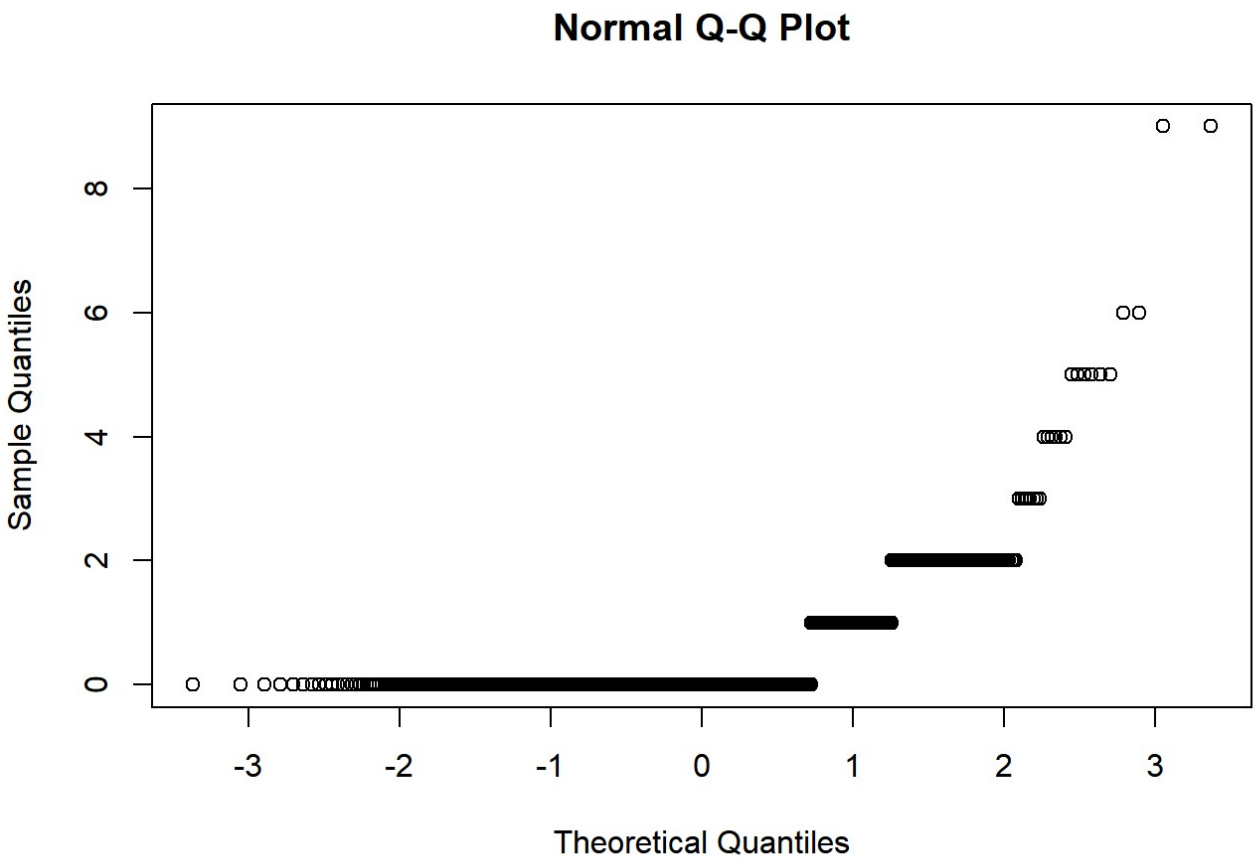
```
hist(Parch)
```



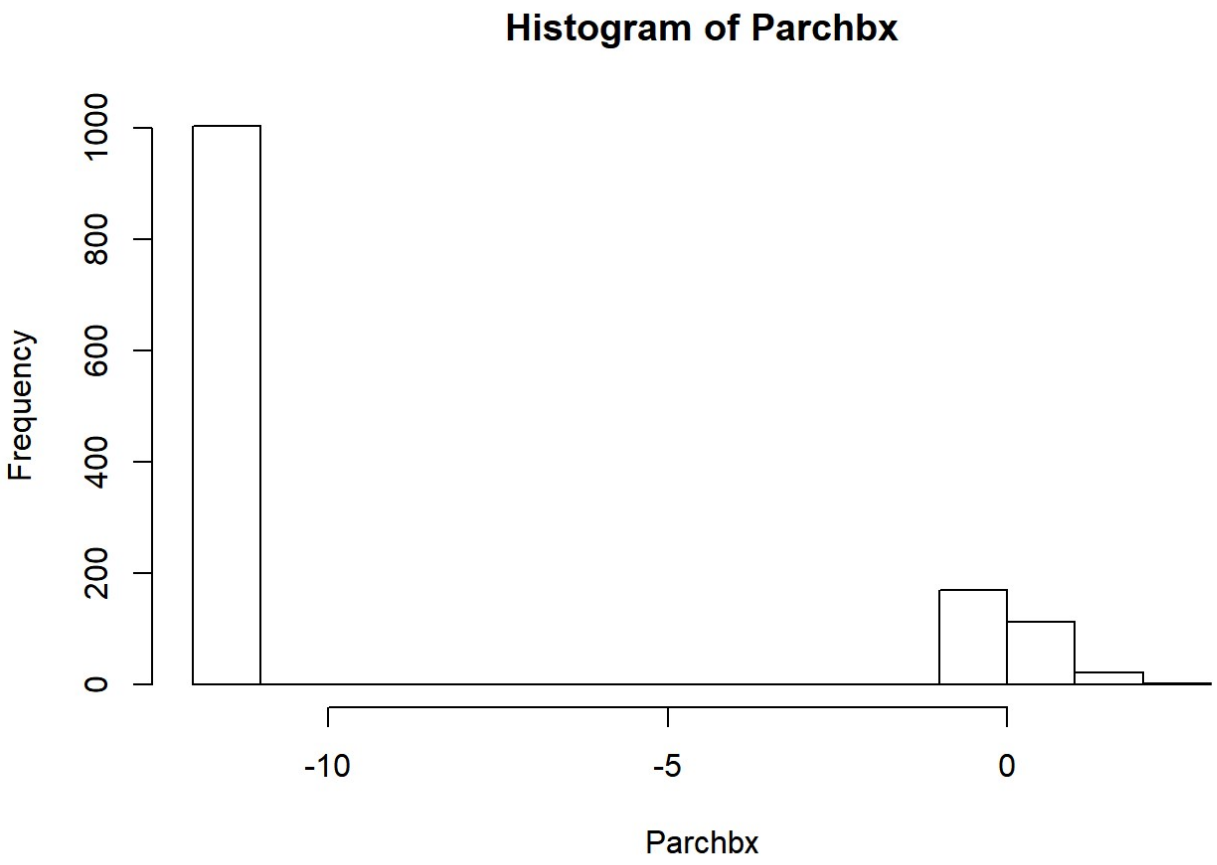

```
shapiro.test(Parch)
```

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

```
qqnorm(Parch)
```



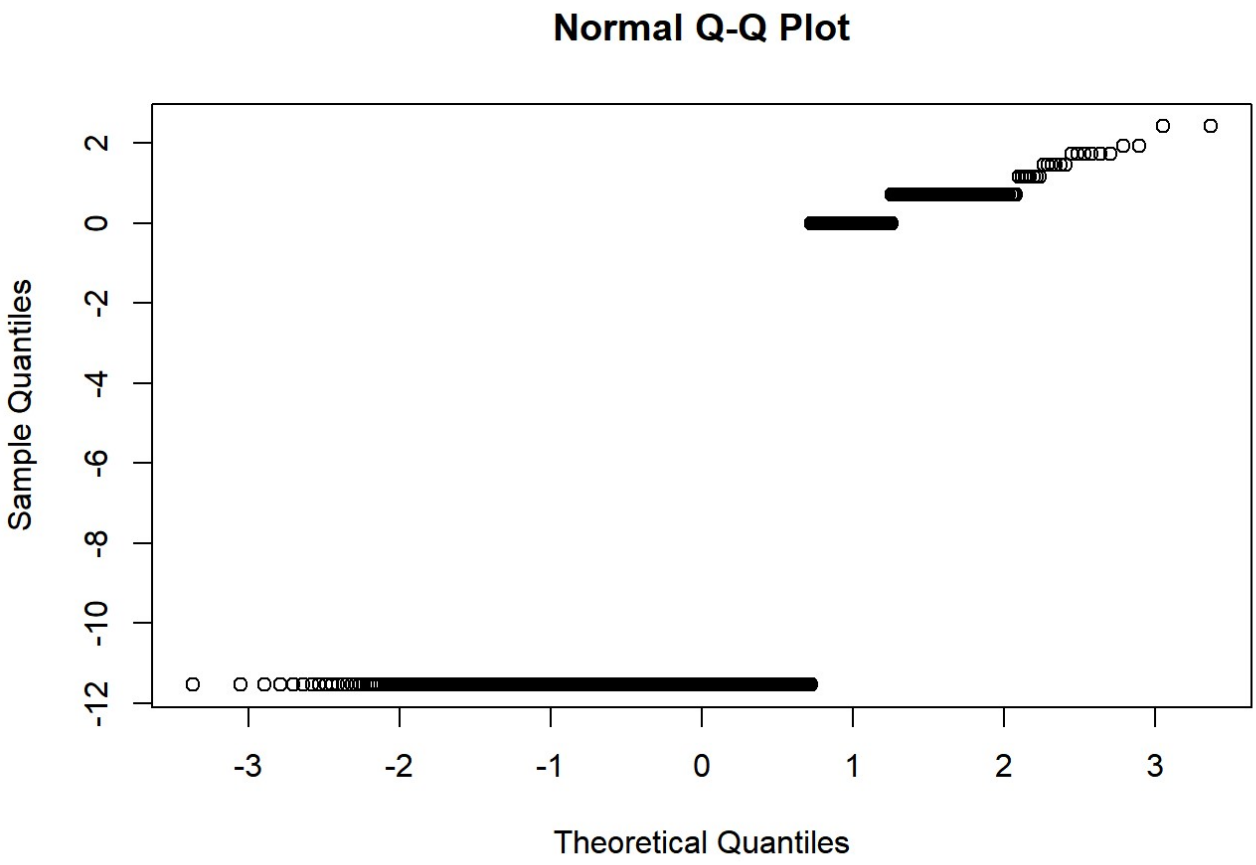

```
Parchbx <- BoxCox(Parch, lambda = BoxCoxLambda(Parch))
hist(Parchbx)
```




```
shapiro.test(Parchbx)
```

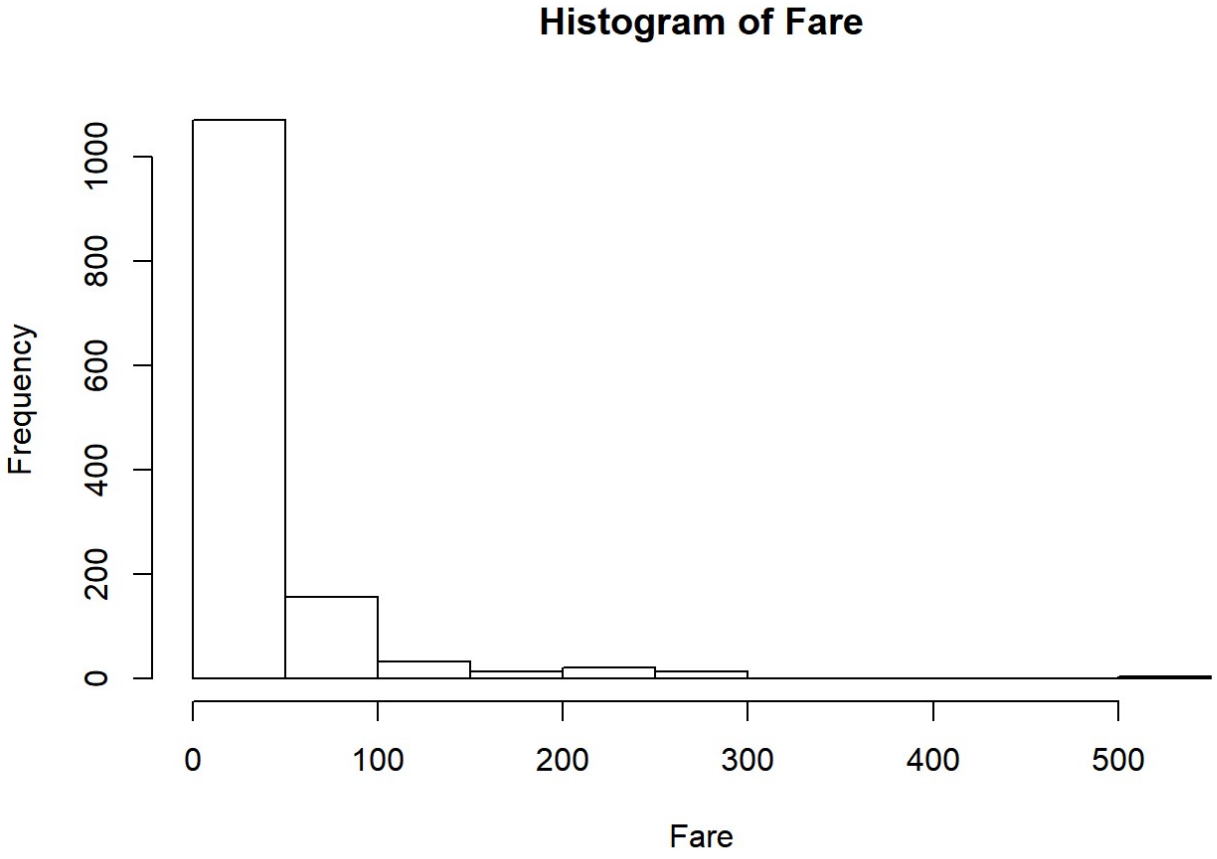
```
##
##  Shapiro-Wilk normality test
##
## data:  Parchbx
## W = 0.53845, p-value < 2.2e-16
```

```
qqnorm(Parchbx)
```



Por último, la tarifa:

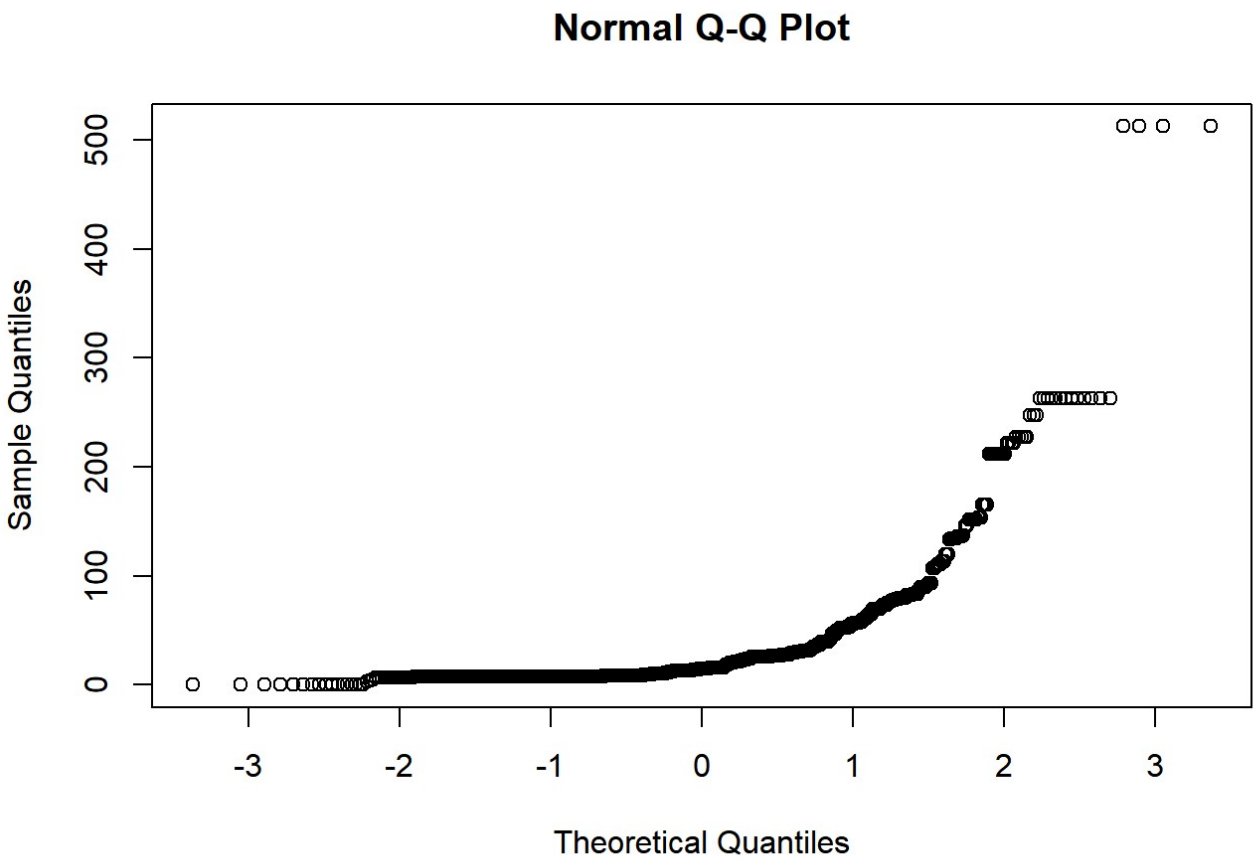
```
hist(Fare)
```



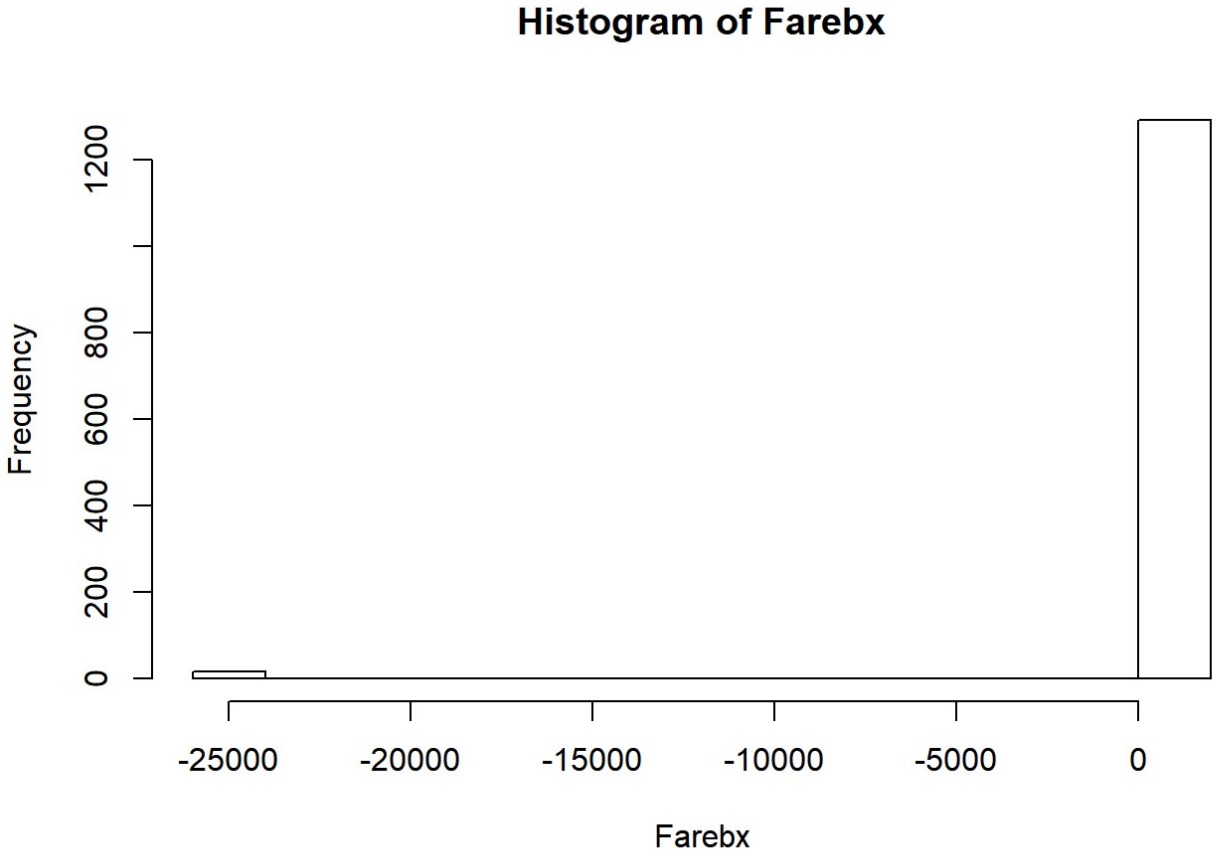

```
shapiro.test(Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Fare
## W = 0.52765, p-value < 2.2e-16
```

```
qqnorm(Fare)
```



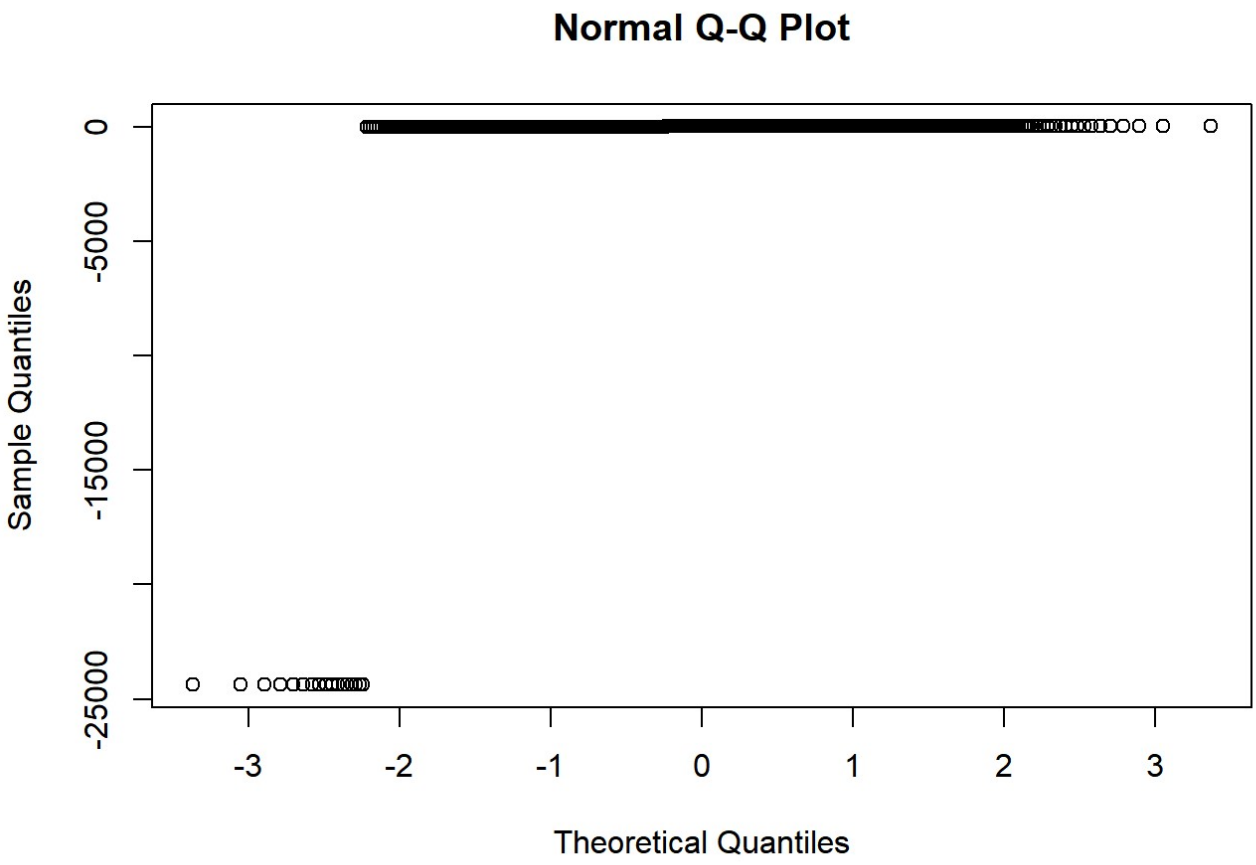

```
Farebx <- BoxCox(Fare, lambda = BoxCoxLambda(Fare))
hist(Farebx)
```




```
shapiro.test(Farebx)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Farebx
## W = 0.087498, p-value < 2.2e-16
```

```
qqnorm(Farebx)
```



Hemos podido comprobar aquí que debemos rechazar la hipótesis de normalidad en todas estas variables, con o sin transformación de Box-Cox, por lo pequeño de sus valores *p*.

4.3. Pruebas estadísticas

Tal y como comentábamos en la sección 4.1, procederemos ahora a realizar las tres pruebas estadísticas que hemos considerado más interesantes.

4.3.1. Discriminación de precios por edad y/o sexo

En esta prueba queremos dilucidar si el viaje del Titanic tenía precios diferentes en razón de la edad o del sexo, dentro de una misma clase de pasaje. Para ello recurriremos al contraste de hipótesis. Como hemos encontrado que *Fare* no es una variable normal, recurriremos al contraste de Wilcoxon.

```
# Contraste en general
wilcox.test(Fare~Sex)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Fare by Sex
## W = 253690, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
# Contraste dentro de la primera clase
wilcox.test(Fare~Sex, subset = Pclass==1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Fare by Sex
## W = 18329, p-value = 6.886e-11
## alternative hypothesis: true location shift is not equal to 0
```

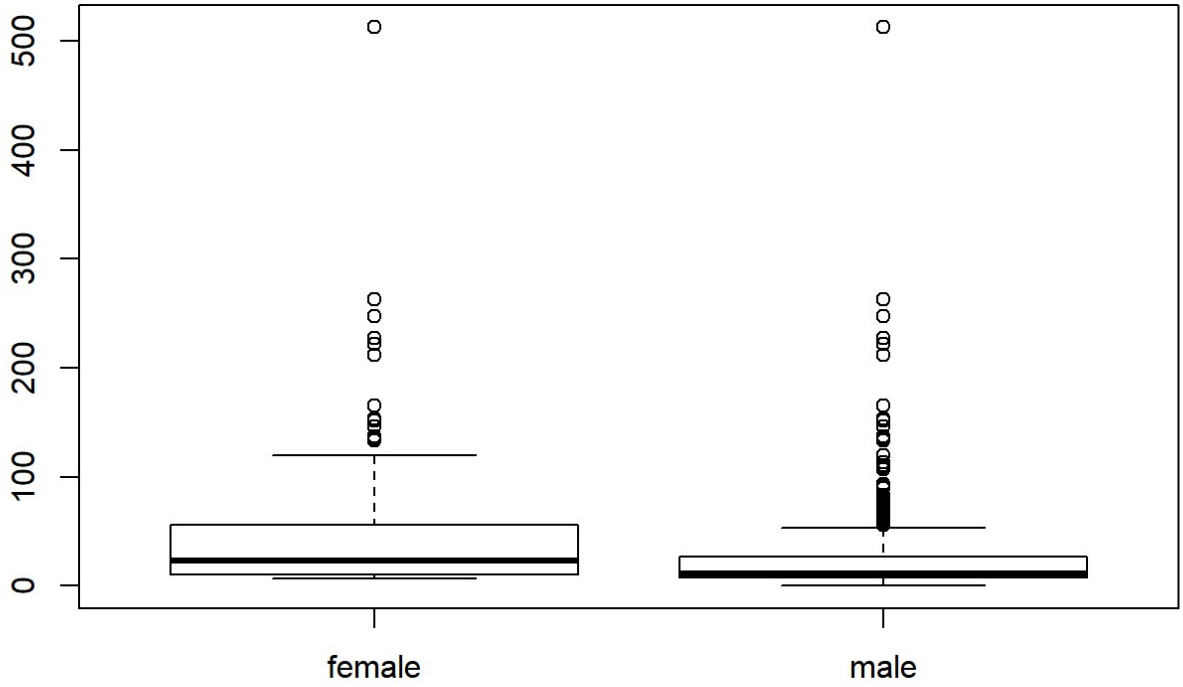
```
# Contraste dentro de la segunda clase
wilcox.test(Fare~Sex, subset = Pclass==2)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Fare by Sex
## W = 11514, p-value = 0.0001379
## alternative hypothesis: true location shift is not equal to 0
```

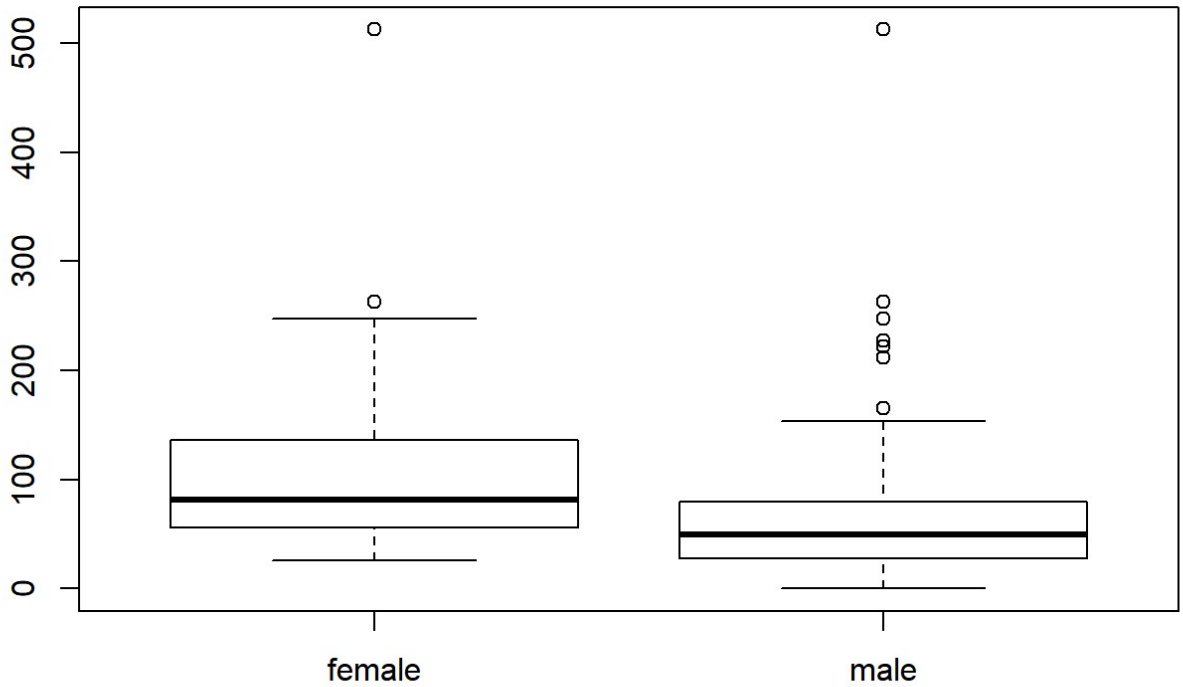
```
# Contraste dentro de la segunda clase
wilcox.test(Fare~Sex, subset = Pclass==3)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Fare by Sex
## W = 66198, p-value = 2.403e-07
## alternative hypothesis: true location shift is not equal to 0
```

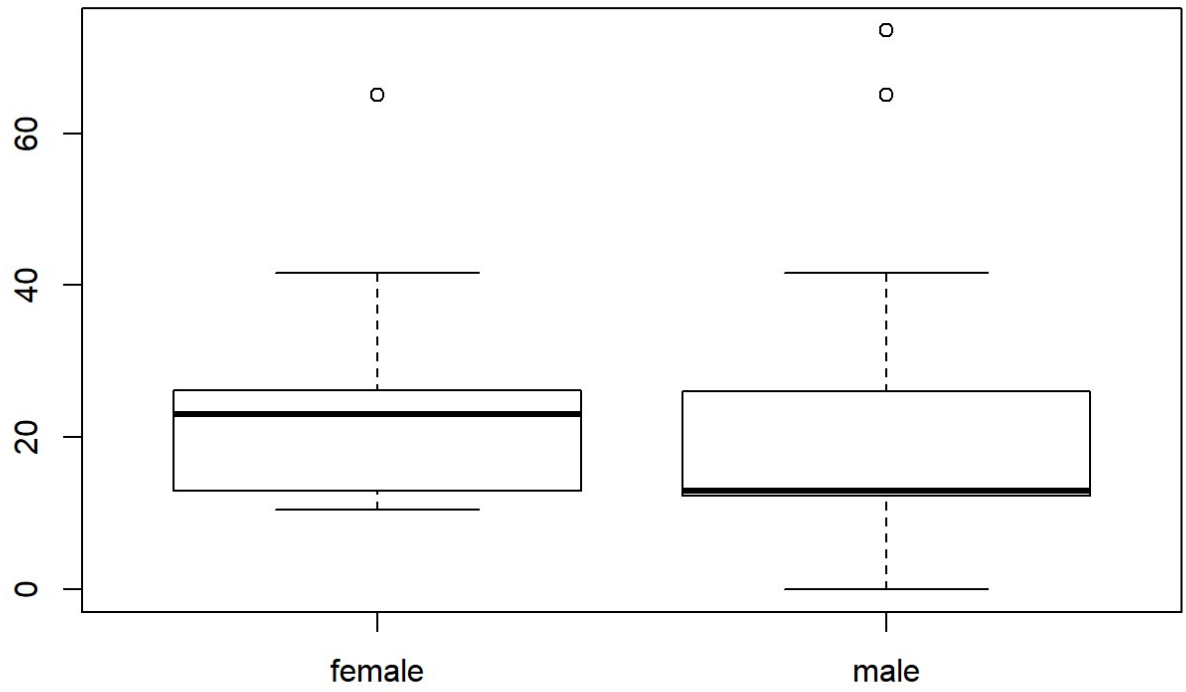
```
# Gráficas
boxplot(Fare~Sex)
```



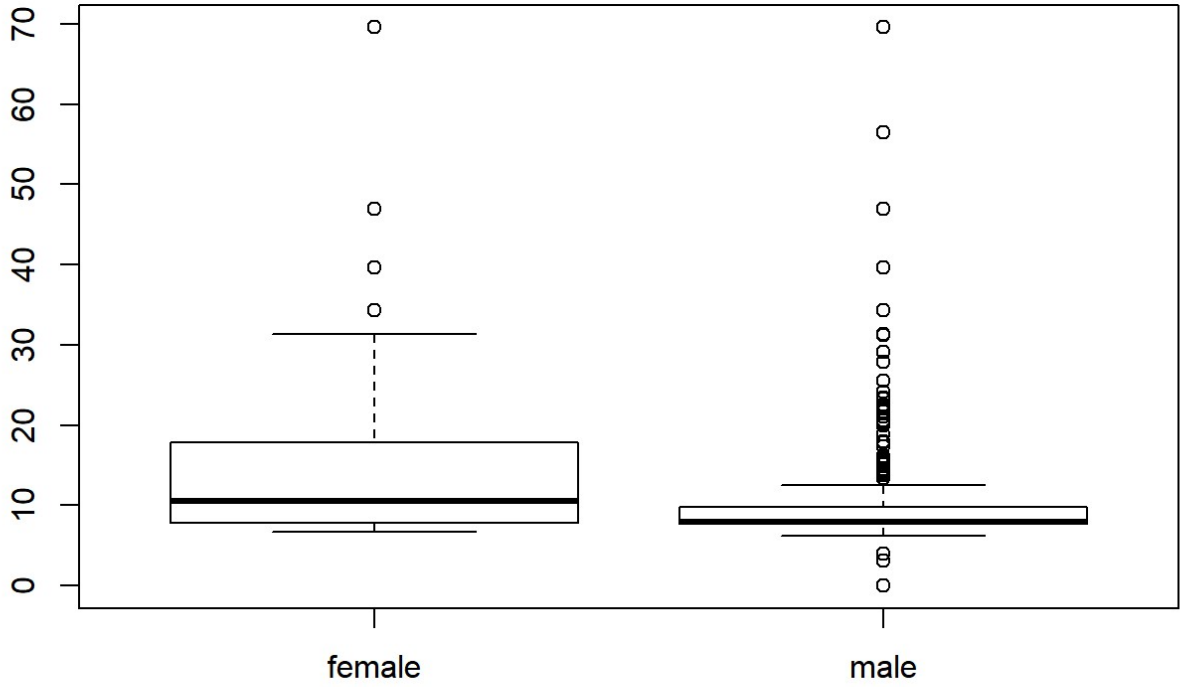

```
boxplot(Fare~Sex, subset = Pclass==1)
```



```
boxplot(Fare~Sex, subset = Pclass==2)
```




```
boxplot(Fare~Sex, subset = Pclass==3)
```



Gráficamente nos encontramos con que las mujeres pagaban más por un billete en el Titanic. Analíticamente, hemos podido comprobar que sí había discriminación de precios, hallándonos en las pruebas de Wilcoxon con valores *p* muy pequeños, que nos hacen rechazar la hipótesis de igualdad.

4.3.2. Modelo de regresión lineal para el precio

Elaboraremos un modelo de regresión lineal que explique el precio de un billete del Titanic a partir de los datos que conocemos. En concreto,lo haremos depender del sexo (acabamos de ver que sí ahbía discriminación), del lugar de embarque y, por supuesto, de la clase.

```
# Reordenamos las variables categóricas
SexR <- relevel(Sex, ref = 'male')
EmbarkedR <- relevel(Embarked, ref = 'S')

# Modelo de regresión lineal
modelo <- lm(Fare ~ Pclass+SexR+EmbarkedR)
summary(modelo)
```

```
##
## Call:
## lm(formula = Fare ~ Pclass + SexR + EmbarkedR)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -68.44 -20.55  -4.19   4.88 428.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  96.85424    3.91955  24.711  < 2e-16 ***
## Pclass      -31.22925    1.48144 -21.080  < 2e-16 ***
## SexRfemale   12.15723    2.46270   4.937 8.98e-07 ***
## EmbarkedRC   18.10825    2.99455   6.047 1.92e-09 ***
## EmbarkedRQ    0.01153    4.14213   0.003    0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.96 on 1304 degrees of freedom
## Multiple R-squared:  0.3445, Adjusted R-squared:  0.3425
## F-statistic: 171.3 on 4 and 1304 DF,  p-value: < 2.2e-16
```

Este modelo lineal sólo nos permite explicar el 34% de las variaciones entre un precio y otro, pero sólo necesitamos saber si la persona es mujer, si va a embarcar en Cherburgo (paga más), y la clase de su pasaje.

4.3.3. Modelo de regresión logística de la supervivencia al desastre

Para la construcción de este modelo debemos prescindir del conjunto de datos de prueba original.

```
# Datos que usaremos
dataglm <- datai[1:891,]
dataglm$Survived <- as.logical(dataglm$Survived)

# Modelo de regresión logística
model1 <- glm(Survived ~ Pclass+Sex+Age+SibSp+Parch+Fare+Embarked, family = 'binom
ial', data = dataglm)
summary(model1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare + Embarked, family = "binomial", data = dataglm)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7743  -0.5933  -0.4020   0.6207   2.5187
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.896313   0.600155   9.825  < 2e-16 ***
## Pclass      -1.250788   0.152266  -8.215  < 2e-16 ***
## Sexmale     -2.684011   0.202390 -13.262  < 2e-16 ***
## Age         -0.049460   0.008201  -6.031 1.63e-09 ***
## SibSp       -0.389889   0.111427  -3.499 0.000467 ***
## Parch       -0.091548   0.121487  -0.754 0.451113
## Fare         0.001508   0.002374   0.635 0.525432
## EmbarkedQ   -0.020669   0.393406  -0.053 0.958100
## EmbarkedS   -0.375343   0.238857  -1.571 0.116087
## ---
## 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:  772.13  on 882  degrees of freedom
## AIC: 790.13
##
## Number of Fisher Scoring iterations: 5
```

```
# Modelo de regresión logística mejorado
model2 <- glm(Survived ~ Pclass+Sex+Age+SibSp, family = 'binomial', data = datagl
m)
summary(model2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = "binomial",
##     data = dataglm)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8309  -0.5985  -0.3873   0.6088   2.5004
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.830057   0.520641  11.198  < 2e-16 ***
## Pclass      -1.315816   0.129235 -10.182  < 2e-16 ***
## Sexmale     -2.698788   0.195833 -13.781  < 2e-16 ***
## Age         -0.050329   0.008143  -6.181 6.38e-10 ***
## SibSp       -0.424870   0.106118  -4.004 6.24e-05 ***
## ---
## 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:  776.87  on 886  degrees of freedom
## AIC: 786.87
##
## Number of Fisher Scoring iterations: 5
```

Tras realizar el modelo hemos decidido mejorarlo quitando las variables que no afectan a la superviviencia a la vista del elevado valor *p* de su contraste individual, como el lugar de embarque, el número de padres e hijos o lo que se ha pagado por el billete. Finalmente, tenemos que ser hombre reduce sensiblemente las probabilidades de sobrevivir a una catástrofe así; ser de clases inferiores también afecta negativamente a la supervivencia; ser muy mayor tampoco ayuda; y, curiosamente, viajar con el cónyuge y/o los hermanos reduce las probabilidades de supervivencia. Los datos parecen sugerir que en situaciones así es mejor no tener a nadie de quién preocuparse.

Predigamos ahora los resultados para la competición de Kaggle:

```
# predicción
datatest <- datai[892:1309,]
predicciones <- predict(model2, datatest, type = 'response')
```

5. Representación visual de los resultados

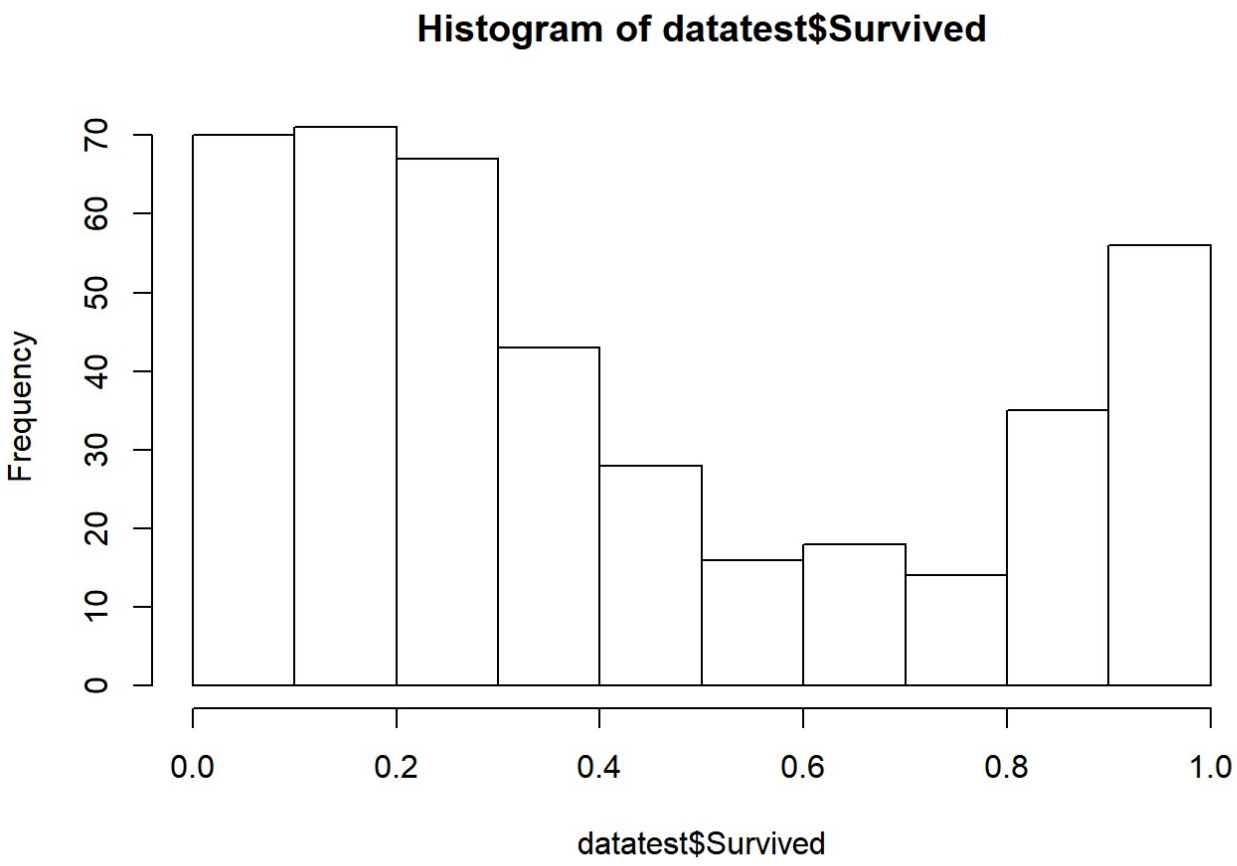
Podemos encontrar la representación visual de los resultados en sus respectivos apartados.

6. Conclusiones

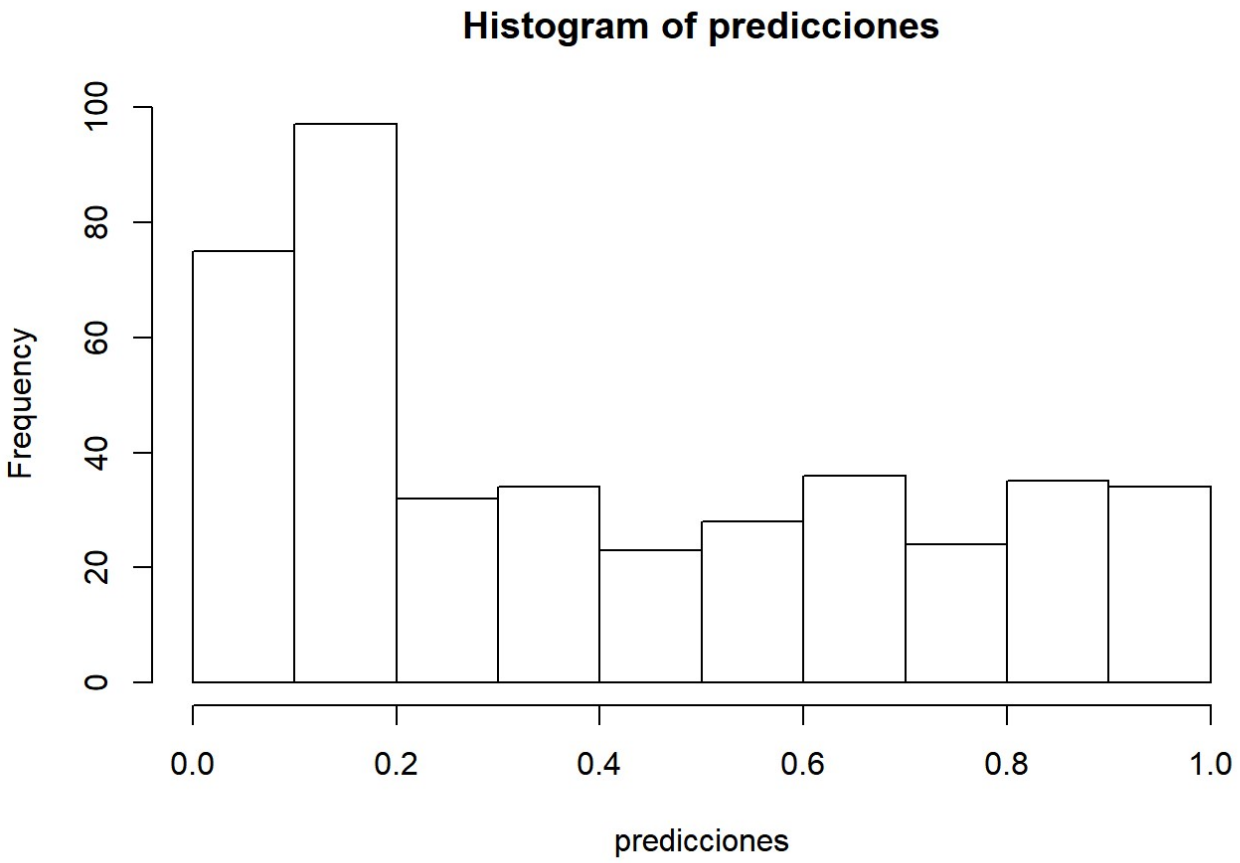
Nos centraremos aquí en la competición de Kaggle, comparando la predicción obtenida en el apartado 4.3.3. con la obtenida en la imputación con missForest.

Enviaremos a Kaggle tanto estas dos predicciones como una tercera basada en la media de las probabilidades calculadas por ambos métodos:

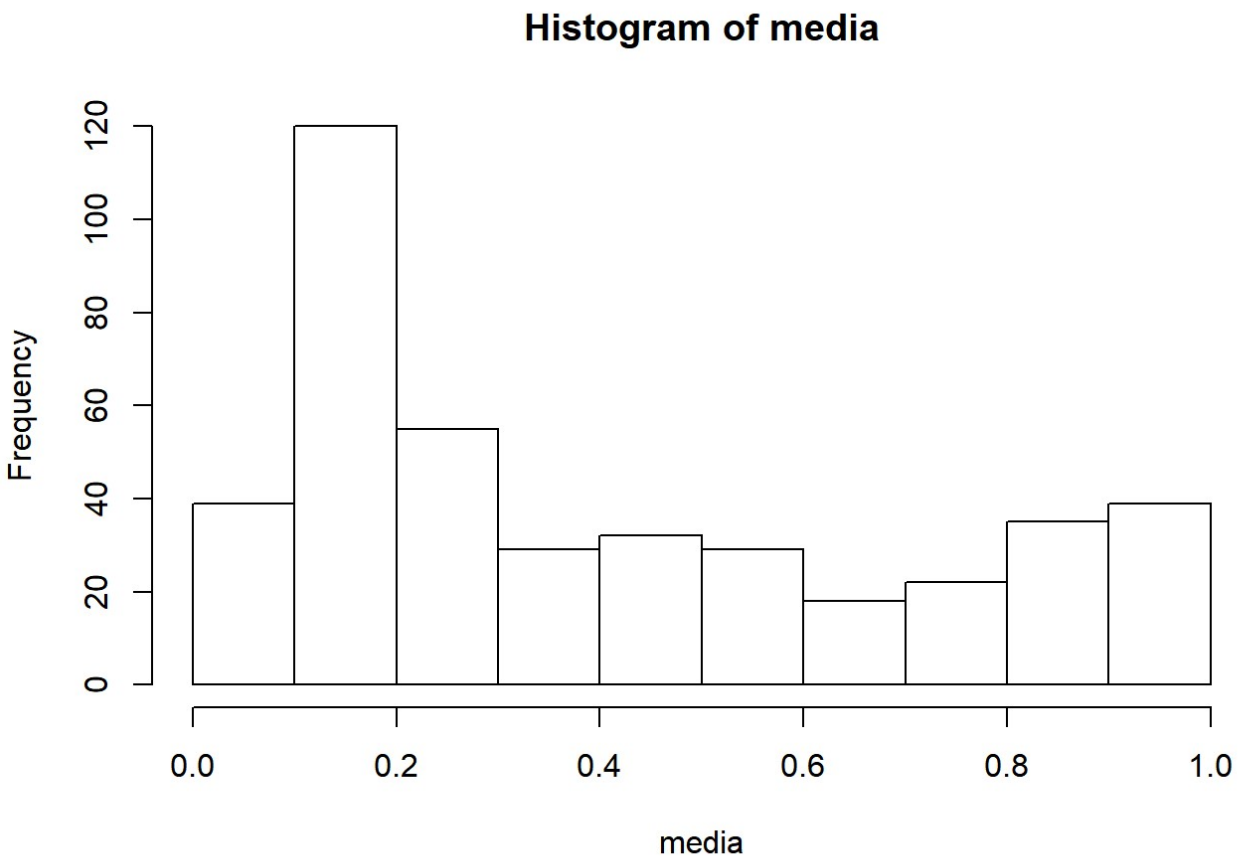
```
# Histogramas
hist(datatest$Survived)
```




```
hist(predicciones)
```




```
media <- (datatest$Survived+predicciones)/2
hist(media)
```




```
# Predicción dicotómica
forestSub <- datatest$Survived>0.5
logitSub <- predicciones>0.5
mediaSub <- media>0.5

# Creación de los ficheros
submission <- read.csv('gender_submission.csv', header = T)
forestSubm <- submission
logitSubm <- submission
mediaSubm <- submission
forestSubm$Survived <- as.integer(forestSub)
logitSubm$Survived <- as.integer(logitSub)
mediaSubm$Survived <- as.integer(mediaSub)
write.csv(forestSubm, file = 'forest.csv', quote = F, row.names= F)
write.csv(logitSubm, file = 'logit.csv', quote = F, row.names= F)
write.csv(mediaSubm, file = 'media.csv', quote = F, row.names= F)
write.csv(datai, file = 'datai.csv', quote = F, row.names= F)
```

```
system('kaggle competitions submit -c titanic -f forest.csv -m "Using missFores
t"')
system('kaggle competitions submit -c titanic -f logit.csv -m "Using a LOGIT mode
l"')
system('kaggle competitions submit -c titanic -f media.csv -m "Using the mean of t
hem"')
```

Una vez en Kaggle, el fichero correspondiente al modelo LOGIT obtuvo una puntuación del 75%, mientras la predicción basada en missForest y la media de ambas alacanzaron una puntuación del 76%.

7. Código

Podemos encontrar el código R utilizado en este documento a lo largo de todo él. Hemos utilizado *R Markdown* para ello.

8. Referencias

Calvo, M., Subirats, L., & Pérez, D. (2019). Introducción a la limpieza y análisis de los datos. Barcelona: UOC.

Kaggle. Titanic: Machine Learning from Disaster. (https://www.kaggle.com/c/titanic (https://www.kaggle.com/c/titanic)) [Consulta: 1 de junio de 2019]

```
kable(data.frame(Contribuciones = c('Investigación previa', 'Redacción de las resp
uestas', 'Desarrollo código'), Firma = c('AFB, JANY', 'AFB, JANY', 'AFB, JANY')))
```

Contribuciones	**Firma**
Investigación previa	AFB, JANY
Redacción de las respuestas	AFB, JANY
Desarrollo código	AFB, JANY