Tipología de Datos - PRA2

Ignacio Fernandez Estebanez

1/5/2021

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

Este documento da respuesta a la práctica 2 de la asignatura *Tipología y Ciclo de Vida de los Datos*. A lo largo de la práctica, se desea afrontar el reto mencionado en el enunciado entre las posibilidades de selección de datasets, de la competición de Kaggle sobre la predicción de los supervivientes del Titanic. https://www.kaggle.com/c/titanic/overview

El problema se enuncia de la siguiente manera: se dispone de un dataset con información de los pasajeros del Titanic dividido en 2 archivos: uno para training y otro para test. Se desea construir un modelo que prediga la supervivencia del pasajero.

A lo largo de este documento, se abordarán las preguntas planteadas en el enunciado de la práctica, para realizar el análisis del dataset.

De acuerdo con la información del reto, se dispone de la siguiente descripción de los datos: https://www.kaggle.com/c/titanic/data

PassengerId: Id del pasajero.

Survived: booleano. Si el pasajero sobrevivió al naufragio.

Pclass: Clase del billete del pasajero.

Sex: Sexo del pasajero.

Age: Edad del pasajero.

SibSp: Número de hermanas y hermanos o conyuges a bordo del barco.

Parch: Número de padres/ hijos a bordo del barco.

Ticket: Ticket number.

Fare: Tarifa.

Cabin: Número de cabina.

Embarked: Puerto de embarco.

Integración y selección de los datos de interés a analizar

El primer paso es cargar el dataset para poder trabajar con los datos y comprobar la descripción que se proporciona. Para ello, se carga el archivo de training descargado previamente de Kaggle.

```
library(readr)
training <- read.csv("~/Downloads/titanic/train.csv", sep = ",")
head(training)
```

```
##   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
```

Se aprecia que en efecto los datos que se disponen coinciden con la descripción de Kaggle. Se dispone de información acerca de 891 pasajeros (más los que estén en la lista de test para validar la predicción). Además se dispone también del nombre del pasajero, aunque no aporta mucho valor estadístico para que problema que se aborda.

A continuación, vamos a analizar los tipos de datos disponibles en el dataset:

```
str(training)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

De acuerdo con la descripción de los datos del enunciado, vamos a realizar algunas transformaciones para facilitar el uso de los datos:

```
# Set Survived as factor
training$Survived <- as.factor(training$Survived)

# Set Class as factor
training$Pclass <- as.factor(training$Pclass)
```

```
# Set Sex as factor
training$Sex <- as.factor(training$Sex)

# Set Embarked as factor
training$Embarked[training$Embarked=='C'] <- 'Cherbourg'
training$Embarked[training$Embarked=='Q'] <- 'Queenstown'
training$Embarked[training$Embarked=='S'] <- 'Southampton'
training$Embarked <- as.factor(training$Embarked)

str(training)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : Factor w/ 2 levels "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     : 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/ 4 levels "","Cherbourg",..: 4 2 4 4 4 3 4 4 4 2 ...
```

Finalmente, vamos a eliminar del dataset las columnas de *Name* y *Ticket* que a priori, no deberían ser relevantes para un análisis estadístico:

```
training <- subset(training, select = - `Name`)
training <- subset(training, select = - `Ticket`)

head(training)
```

```
##   PassengerId Survived Pclass    Sex Age SibSp Parch    Fare Cabin    Embarked
## 1           1        0      3   male  22     1     0  7.2500       Southampton
## 2           2        1      1 female  38     1     0 71.2833   C85   Cherbourg
## 3           3        1      3 female  26     0     0  7.9250       Southampton
## 4           4        1      1 female  35     1     0 53.1000  C123 Southampton
## 5           5        0      3   male  35     0     0  8.0500       Southampton
## 6           6        0      3   male  NA     0     0  8.4583        Queenstown
```

Una vez hemos seleccionado y adaptado los datos con los que queremos trabajar, ya podemos pasar al punto de limpieza de datos.

Limpieza de los datos.

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

A continuación vamos a realizar la limpieza de los datos. En primer lugar, vamos a comprobar qué atributos disponen de valores nulos:

```r
summary(training)
```

```
##   PassengerId    Survived Pclass      Sex           Age            SibSp
##  Min.   :  1.0   0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000
##  1st Qu.:223.5   1:342    2:184   male  :577   1st Qu.:20.12   1st Qu.:0.000
##  Median :446.0            3:491                Median :28.00   Median :0.000
##  Mean   :446.0                                 Mean   :29.70   Mean   :0.523
##  3rd Qu.:668.5                                 3rd Qu.:38.00   3rd Qu.:1.000
##  Max.   :891.0                                 Max.   :80.00   Max.   :8.000
##                                                NA's   :177
##      Parch             Fare           Cabin                 Embarked
##  Min.   :0.0000   Min.   :  0.00   Length:891                    :  2
##  1st Qu.:0.0000   1st Qu.:  7.91   Class :character   Cherbourg  :168
##  Median :0.0000   Median : 14.45   Mode  :character   Queenstown : 77
##  Mean   :0.3816   Mean   : 32.20                      Southampton:644
##  3rd Qu.:0.0000   3rd Qu.: 31.00
##  Max.   :6.0000   Max.   :512.33
##
```

Aparentemente se disponen de valores nulos en los siguientes campos:

- **Age**: hay 177 pasajeros de los que desconocemos su edad.
  - **Embarked**: hay 2 pasajeros de los que no se dispone información sobre el puerto de embarque.

Adicionalmente, vamos a comprobar en detalle los valores de cabin:

```r
table(training$Cabin)
```

```
##
##                         A10             A14             A16             A19
##             687             1               1               1               1
##             A20             A23             A24             A26             A31
##               1             1               1               1               1
##             A32             A34             A36              A5              A6
##               1             1               1               1               1
##              A7            B101            B102             B18             B19
##               1             1               1               2               1
##             B20             B22             B28              B3             B30
##               2             2               2               1               1
##             B35             B37             B38             B39              B4
##               2             1               1               1               1
##             B41             B42             B49              B5             B50
##               1             1               2               2               1
##     B51 B53 B55 B57 B59 B63 B66         B58 B60             B69             B71
##               2             2               2               1               1
##             B73             B77             B78             B79             B80
##               1             2               1               1               1
##         B82 B84             B86             B94         B96 B98            C101
##               1             1               1               4               1
##            C103            C104            C106            C110            C111
##               1             1               1               1               1
```

```
##        C118        C123        C124        C125        C126
##           1           2           2           2           2
##        C128        C148          C2    C22 C26 C23 C25 C27
##           1           1           2           3           4
##         C30         C32         C45         C46         C47
##           1           1           1           1           1
##         C49         C50         C52         C54     C62 C64
##           1           1           2           1           1
##         C65         C68          C7         C70         C78
##           2           2           1           1           2
##         C82         C83         C85         C86         C87
##           1           2           1           1           1
##         C90         C91         C92         C93         C95
##           1           1           2           2           1
##         C99           D     D10 D12         D11         D15
##           1           3           1           1           1
##         D17         D19         D20         D21         D26
##           2           1           2           1           2
##         D28         D30         D33         D35         D36
##           1           1           2           2           2
##         D37         D45         D46         D47         D48
##           1           1           1           1           1
##         D49         D50         D56          D6          D7
##           1           1           1           1           1
##          D9         E10        E101         E12        E121
##           1           1           3           1           2
##         E17         E24         E25         E31         E33
##           1           2           2           1           2
##         E34         E36         E38         E40         E44
##           1           1           1           1           2
##         E46         E49         E50         E58         E63
##           1           1           1           1           1
##         E67         E68         E77          E8       F E69
##           2           1           1           2           1
##      F G63       F G73          F2         F33         F38
##           1           2           3           3           1
##          F4          G6           T
##           2           4           1
```

Vemos que hay 687 pasajeros de los que no se dispone de información sobre la cabina.

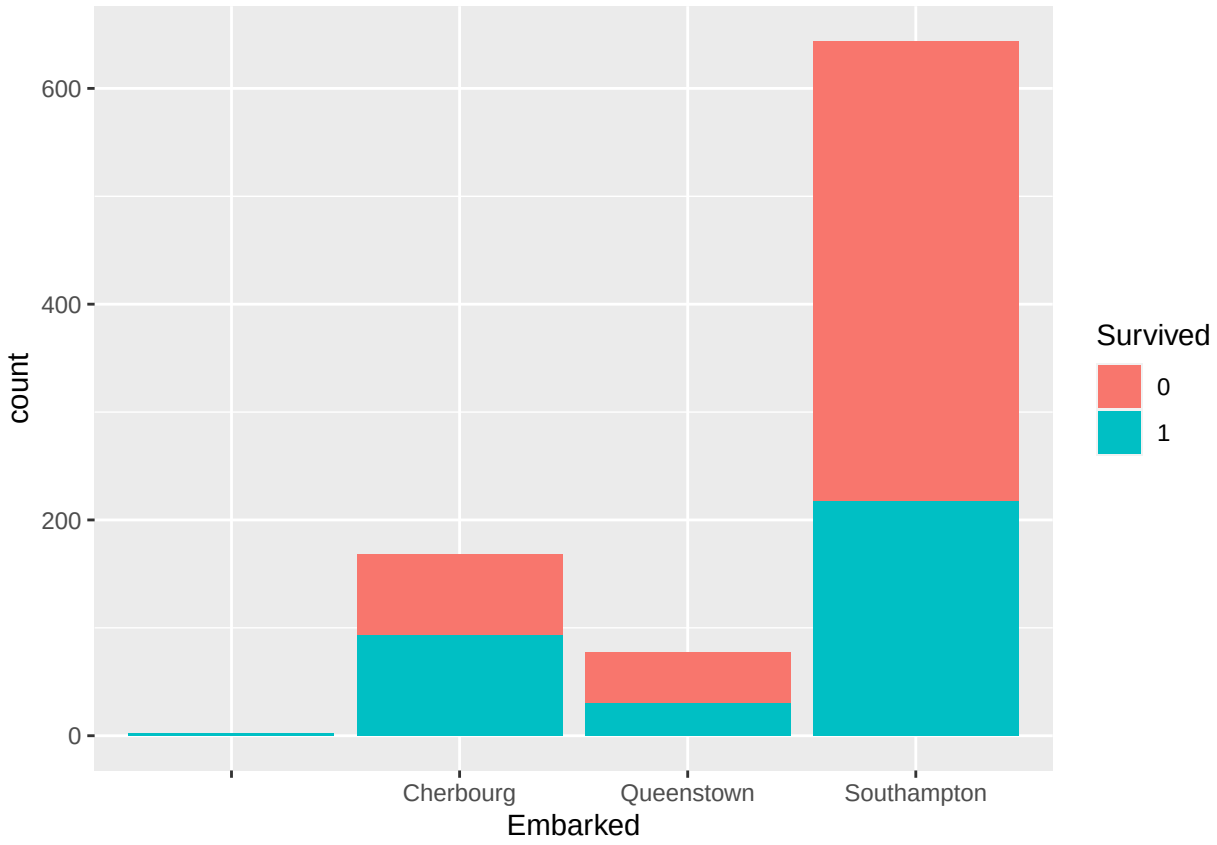
Embarked A los pasajeros que se desconoce su puerto de embarque, se les va a asignar el puerto con mayor afluencia. Si revisamos los pasajeros embarcados en cada puerto:

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
ggplot(data=training, aes(x=Embarked,fill=Survived))+geom_bar()
```



Vemos que el puerto con más afluencia fue el de Southampton. Ese será el que imputaremos a estos viajeros:

```
# Set empty Embarked to Southampton.
training$Embarked[training$Embarked==""] <- "Southampton"

table(training$Embarked)
```

```
##
##              Cherbourg  Queenstown Southampton
##           0         168          77         646
```

Age Para el caso de los datos que no se disponen sobre la edad, vamos a hacer una imputación de la mediana de la edad en función del grupo al que pertenezcan por clase y sexo.

```
ages <- filter(training, !is.na(training$Age))

# Due to issues with knit, I split the calculation and manually place the median for the imputation
#median_1_male <- median(ages$Age[ages$Pclass==1 & ages$Sex=="male"])
#median_2_male<- median(ages$Age[ages$Pclass==2 & ages$Sex=="male"])
#median_3_male <- median(ages$Age[ages$Pclass==3 & ages$Sex=="male"])
#median_1_female <- median(ages$Age[ages$Pclass==1 & ages$Sex=="female"])
#median_2_female <- median(ages$Age[ages$Pclass==2 & ages$Sex=="female"])
```

```
#median_3_female <- median(ages$Age[ages$Pclass==3 & ages$Sex=="female"])

training$Age[is.na(training$Age) & training$Pclass==1 & training$Sex=="male"] <- 40
training$Age[is.na(training$Age) & training$Pclass==2 & training$Sex=="male"] <- 30
training$Age[is.na(training$Age) & training$Pclass==3 & training$Sex=="male"] <- 25
training$Age[is.na(training$Age) & training$Pclass==1 & training$Sex=="female"] <- 35
training$Age[is.na(training$Age) & training$Pclass==2 & training$Sex=="female"] <- 28
training$Age[is.na(training$Age) & training$Pclass==3 & training$Sex=="female"] <- 21.5

summary(training$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   21.50   26.00   29.11   36.00   80.00
```

Cabin En el caso de los datos de la cabina, hay demasiados valores ausentes. Son 687 sobre 891 observaciones, por lo que realizar una imputación nos podría conducir a demasiado error. En este caso dejaremos el atributo como está y **no lo consideraremos en el dataset**.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
training <- select(training, x=-Cabin)
```

Identificación y tratamiento de valores extremos

A continuación vamos a hacer un análisis de outliers en los datos que se disponen. Los outliers son valores que toman los datos que se salen significativamente del rango de la distribución. Este análisis se aplica a los valores numéricos. Podemos analizarlo de diferentes formas, pero una de las más comunes es el análisis en boxplot donde además se visualizan fácilmente los outliers de forma gráfica.

En el dataset se disponen de los siguientes datos numéricos:

PassengerId

Age

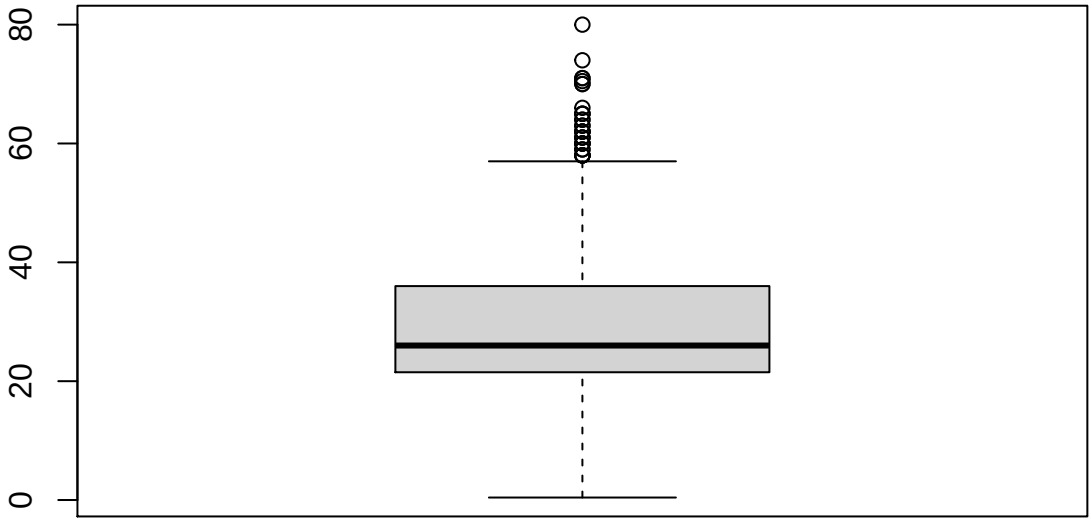
SibSp

Parch

Fare

En el caso de PassengerId, no nos interesan los outliers porque no es más que el identificador del pasajero. Analicemos el resto de datos. Empezamos por Age:

```
boxplot(training$Age)
```




```
boxplot.stats(training$Age)
```

```
## $stats
## [1]  0.42 21.50 26.00 36.00 57.00
##
## $n
## [1] 891
##
## $conf
## [1] 25.23249 26.76751
##
## $out
##  [1] 58.0 66.0 65.0 59.0 71.0 70.5 61.0 58.0 59.0 62.0 58.0 63.0 65.0 61.0 60.0
## [16] 64.0 65.0 63.0 58.0 71.0 64.0 62.0 62.0 60.0 61.0 80.0 58.0 70.0 60.0 60.0
## [31] 70.0 62.0 74.0
```

Vemos que el box plot marca el valor máximo que considera dentro del rango en 57. Este valor se corresponde al tercer cuartil más 1.5 veces el IQR. A partir de este valor, considera los valores outliers.

Vemos que hay 24 pasajeros con edades comprendidas entre 58 y 80 que se consideran outliers desde el punto de vista estadístico. Sin embargo, en 1912 era perfectamente plausible que hubiera en el barco personas en este rango de edad. Por lo tanto, *no se va a hacer ninguna imputación de los outliers*. No obstante, no se considerarán estos valores atípicos a la hora de obtener medidas estadísticas de tendencia central y desviación,

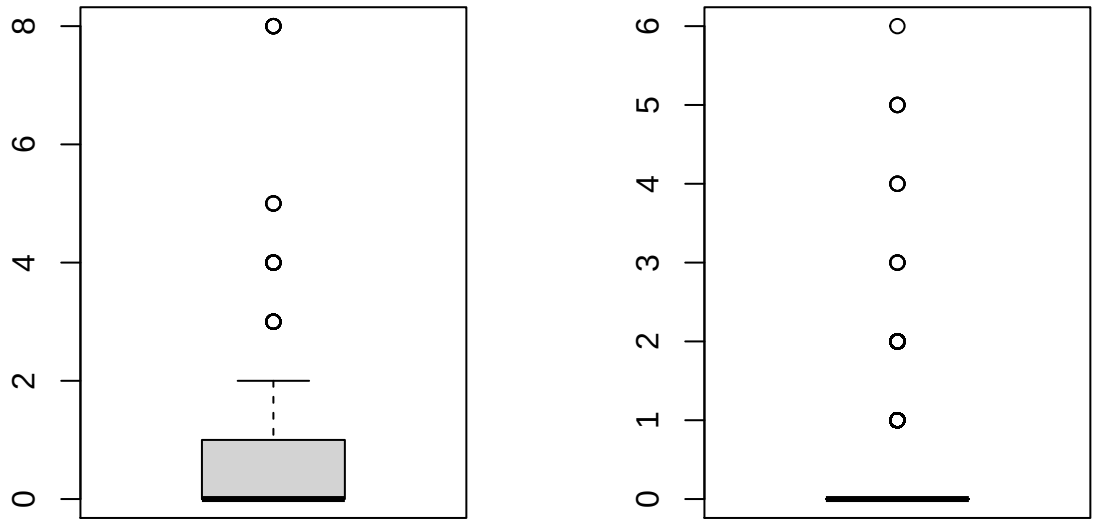
para que no ofrezcan una visión errónea de la distribución. Sino que utilizaremos otras alternativas como, por ejemplo, la media recortada.

A continuación analizamos los valores de SibSp y Parch:

```
par(mfrow=c(1,2))
boxplot(training$SibSp)
boxplot.stats(training$SibSp)
```

```
## $stats
## [1] 0 0 0 1 2
##
## $n
## [1] 891
##
## $conf
## [1] -0.05293199  0.05293199
##
## $out
##  [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3 5 4 3
## [39] 4 8 4 3 4 8 4 8
```

```
boxplot(training$Parch)
```

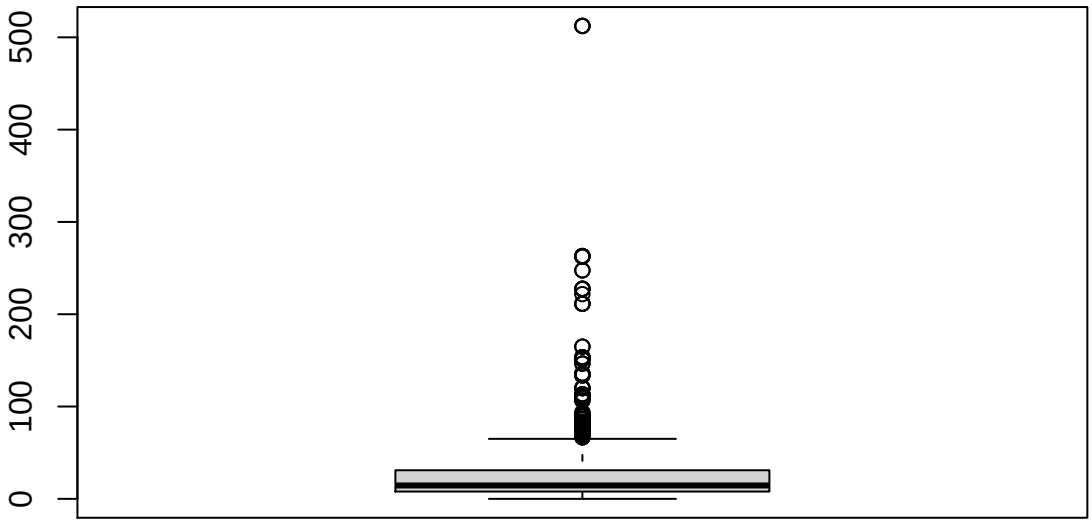


En el caso de SibSp, vemos que lo normal es que los pasajeros viajaran con entre 0 y 2 acompañantes de tipo hermano/hermana esposo/esposa. Vemos que hay algunos casos con 3, 4, 5 y 8 acompañantes de este

tipo. Aunque sea raro si que es posible, por lo que igual que en el caso anterior, no ejecutaremos ninguna imputación en estos casos, pero sí que lo consideraremos de cara al cálculo de medidas de tendencia central.

Finalmente, analizamos la variable Fare:

```
boxplot(training$Fare)
```

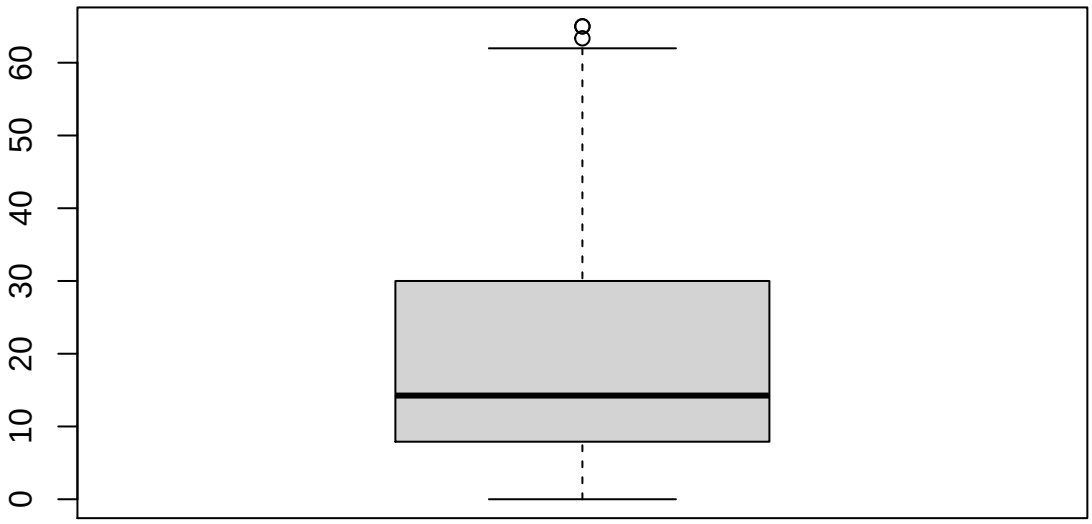



```
# Store the first outlier
boxplot <- boxplot.stats(training$Fare)
limitFare <- min(boxplot$out)
```

Se aprecia que el precio de los billetes se mueve en un rango entre 8 y 31 (supongo que libras o dólares). Sin embargo, hay billetes que ascienden hasta los 500.

En este caso sí que se va a aplicar una imputación de valor sobre los outliers. Lo que se va a hacer es aplicar en cada caso la mediana del precio del billete para cada clase.

```
training$Fare[training$Pclass==1 & training$Fare>=limitFare] <- median(training$Fare[training$Pclass==1
training$Fare[training$Pclass==2 & training$Fare>=limitFare] <- median(training$Fare[training$Pclass==2
training$Fare[training$Pclass==3 & training$Fare>=limitFare] <- median(training$Fare[training$Pclass==3

boxplot(training$Fare)
```



Análisis de los datos.

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

En primer lugar, se seleccionan los datos que se quieren analizar/comparar. En este ejercicio, el objetivo es conseguir un modelo que sea capaz de predecir si un pasajero a bordo del Titanic sobrevive al naufragio o no. Por lo tanto, la división principal que se hace del dataset es en dos clases definidas por el atributo *Survived*.

Para obtener este modelo, se van a analizar los siguientes pasos:

Comprobación de la normalidad y homogeneidad de los datos.

Análisis estadístico para comparar los grupos de datos.

Comprobación de la normalidad y homogeneidad de la varianza

Tenemos dos tipos de variables en el dataset, las categóricas, donde se englobal Survided, Pclass, Sex y Embarked; y las continuas, donde se encuentran Age, SibSp, Parch y Fare.

La comprobación de la normalidad es necesaria para poder hacer análisis posteriores como por ejemplo el contraste de hipótesis. Para las variables contínuas puede hacerse con las pruebas de Kolmogorov-Smirnov y de Shapiro-Wilk. El método de Shapiro-Wilk se considera más robusto, por lo que será el que utilizaremos.

Age

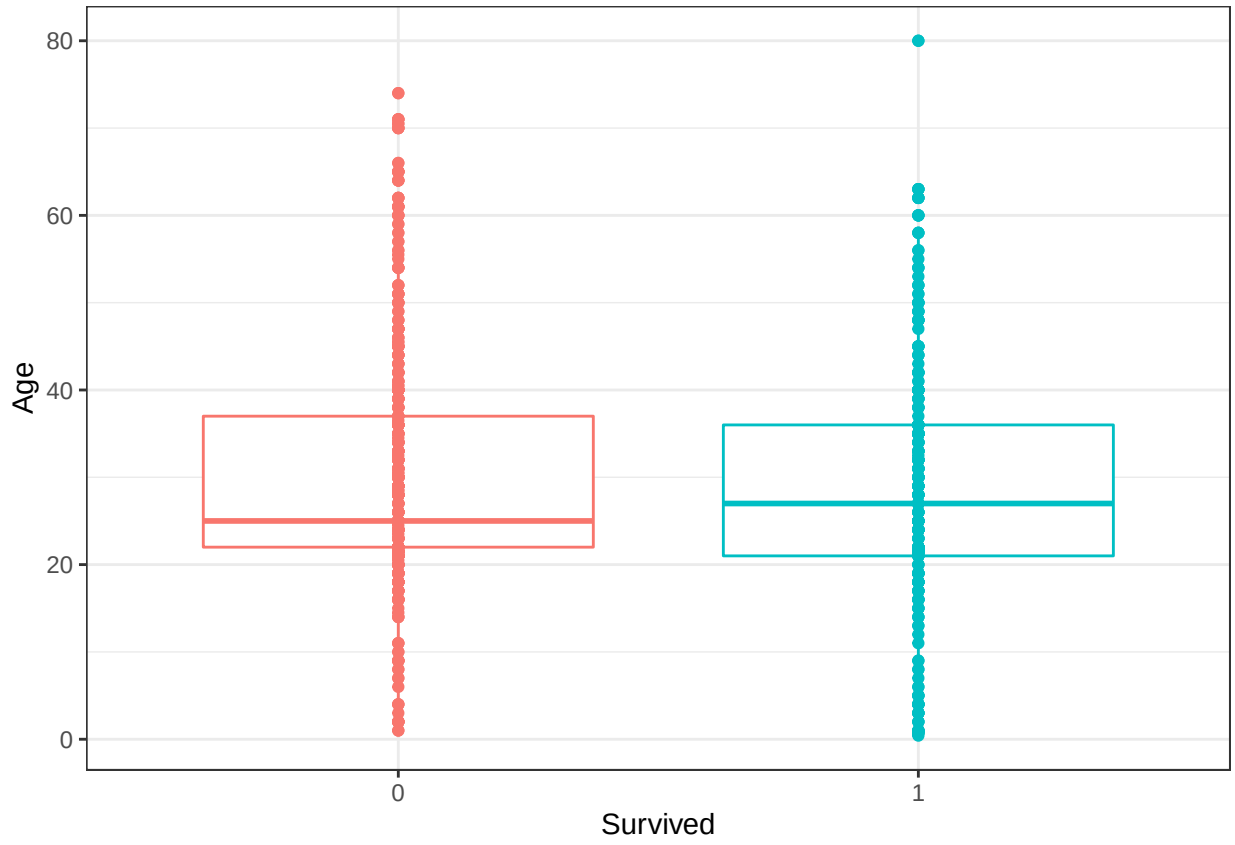
```r
# Convert to numeric for the Shapiro test
shapiro.test(training$Age[training$Survived==0])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$Age[training$Survived == 0]
## W = 0.94026, p-value = 4.791e-14
```

```r
shapiro.test(training$Age[training$Survived==1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$Age[training$Survived == 1]
## W = 0.98016, p-value = 0.0001169
```

```r
# Plot data
ggplot(data = training, aes(x = Survived, y = Age, colour = Survived)) +
  geom_boxplot() +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none")
```



```r
# Validate with Kolmogorov-Smirnov
ks.test(training$Age[training$Survived==1], pnorm, mean(training$Age[training$Survived==1]), sd(trainin
```

```
## Warning in ks.test(training$Age[training$Survived == 1], pnorm,
## mean(training$Age[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  training$Age[training$Survived == 1]
## D = 0.066178, p-value = 0.1
## alternative hypothesis: two-sided
```

```r
ks.test(training$Age[training$Survived==0], pnorm, mean(training$Age[training$Survived==0]), sd(trainin
```

```
## Warning in ks.test(training$Age[training$Survived == 0], pnorm,
## mean(training$Age[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  training$Age[training$Survived == 0]
## D = 0.15236, p-value = 1.705e-11
## alternative hypothesis: two-sided
```

Se concluye por la dos pruebas que la distribución no es normal, dado que el nivel de significancia p está muy por debajo de 0.05. La prueba de Kolmogorov-Smirnov confirma los resultados de la distribución, aunque en este caso el grupo de supervivientes sí que seguiría una distribución normal en cuanto a la edad.

Si nos fijamos en la comparativa de boxplot, vemos que los rangos de edad en ambos grupos son muy parecidos, aunque en el caso de los supervivientes, la mediana queda ligeramente por encima.

Finalmente, vamos a comprobar la homogeneidad de la varianza para los dos grupos. En este caso, debido a la falta de normalidad, se elige aplicar el test de Barlett, que es menos sensible a esta falta (fuente: https://rpubs.com/Joaquin_AR/218466).

```r
bartlett.test(list(training$Age[training$Survived==0],training$Age[training$Survived==1]))
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  list(training$Age[training$Survived == 0], training$Age[training$Survived == 1])
## Bartlett's K-squared = 3.3636, df = 1, p-value = 0.06665
```

El nivel de significancia p-value nos sale superior a 0.05, por lo que aceptamos que la hipótesis de que **ambas poblaciones tienen varianzas semejantes** es correcta, como confirma el boxplot anterior.

SibSp

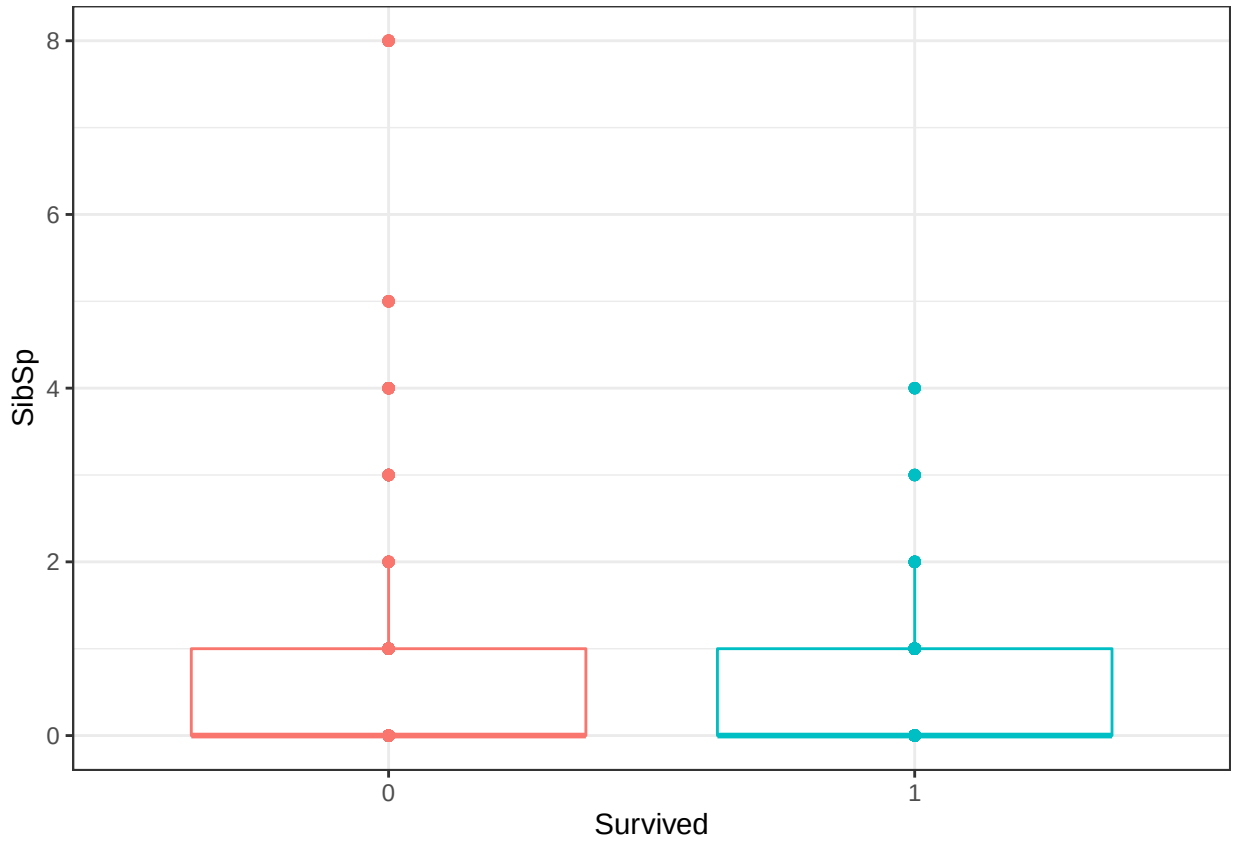
```r
# Convert to numeric for the Shapiro test
shapiro.test(training$SibSp[training$Survived==0])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$SibSp[training$Survived == 0]
## W = 0.48418, p-value < 2.2e-16
```

```r
shapiro.test(training$SibSp[training$Survived==1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$SibSp[training$Survived == 1]
## W = 0.65477, p-value < 2.2e-16
```

```r
# Plot data
ggplot(data = training, aes(x = Survived, y = SibSp, colour = Survived)) +
  geom_boxplot() +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none")
```



```r
# Validate with Kolmogorov-Smirnov
ks.test(training$SibSp[training$Survived==1], pnorm, mean(training$SibSp[training$Survived==1]), sd(tra
```

```
## Warning in ks.test(training$SibSp[training$Survived == 1], pnorm,
## mean(training$SibSp[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  training$SibSp[training$Survived == 1]
## D = 0.36209, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```r
ks.test(training$SibSp[training$Survived==0], pnorm, mean(training$SibSp[training$Survived==0]), sd(tra
```

```
## Warning in ks.test(training$SibSp[training$Survived == 0], pnorm,
## mean(training$SibSp[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  training$SibSp[training$Survived == 0]
## D = 0.39128, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Nuevamente, obtenemos valores de significancia muy inferiores a 0.05, por lo que la interpretación es que la distribución no es normal. Aplicamos nuevamente Barrett para la prueba de homogeneidad de la varianza:

```r
bartlett.test(list(training$SibSp[training$Survived==0],training$SibSp[training$Survived==1]))
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  list(training$SibSp[training$Survived == 0], training$SibSp[training$Survived == 1])
## Bartlett's K-squared = 130.73, df = 1, p-value < 2.2e-16
```

En este caso, obtenemos un nivel de significancia muy por debajo de 0.05, por lo que no se puede afirmar que ambos grupos tengan varianzas semejantes.

Parch

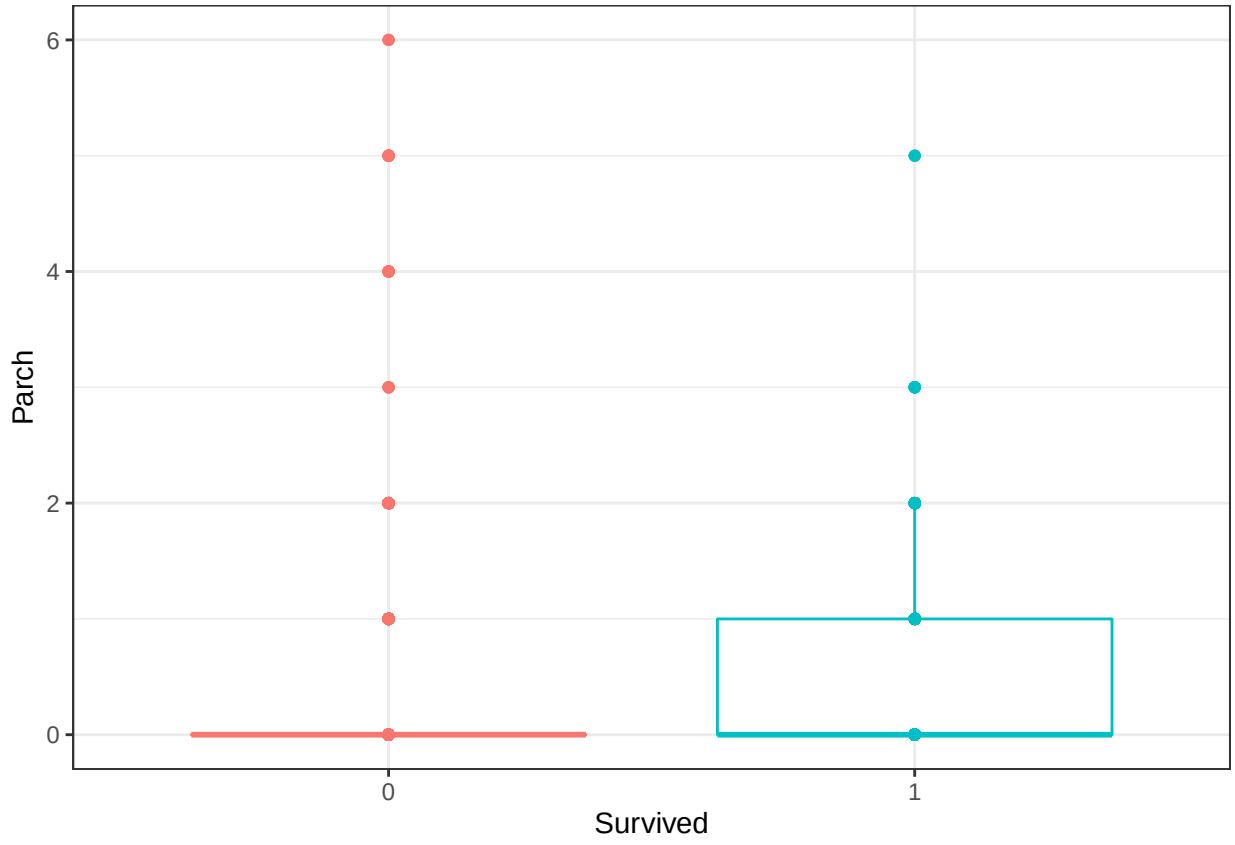
```r
# Convert to numeric for the Shapiro test
shapiro.test(training$Parch[training$Survived==0])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$Parch[training$Survived == 0]
## W = 0.45882, p-value < 2.2e-16
```

```r
shapiro.test(training$Parch[training$Survived==1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$Parch[training$Survived == 1]
## W = 0.63887, p-value < 2.2e-16
```

```r
# Plot data
ggplot(data = training, aes(x = Survived, y = Parch, colour = Survived)) +
  geom_boxplot() +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none")
```




```r
# Validate with Kolmogorov-Smirnov
ks.test(training$Parch[training$Survived==1], pnorm, mean(training$Parch[training$Survived==1]), sd(tra
```

```
## Warning in ks.test(training$Parch[training$Survived == 1], pnorm,
## mean(training$Parch[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
```

```
##
## data:  training$Parch[training$Survived == 1]
## D = 0.40785, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ks.test(training$Parch[training$Survived==0], pnorm, mean(training$Parch[training$Survived==0]), sd(tra
```

```
## Warning in ks.test(training$Parch[training$Survived == 0], pnorm,
## mean(training$Parch[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  training$Parch[training$Survived == 0]
## D = 0.46618, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

De nuevo, obtenemos niveles de significancia muy por debajo del nivel para aceptar la hipótesis, por lo que la distribución no es normal. En este caso, se aprecia una diferencia significativa además en el boxplot de ambos grupos. Se aprecia, que la distribución entre los participantes en el grupo que se salvó, oscila entre 0-1, mientras que en el grupo de los que falleció se sitúa en el cero. Esto daría sentido a la premisa de que salvaron tantos niños y niñas como pudieron, y por ello, la cantidad de progenitores o hijos a bordo entre los pasajeros que se salvaron sería superior a uno.

Aplicamos nuevamente la comprobación de Barret:

```
bartlett.test(list(training$Parch[training$Survived==0],training$Parch[training$Survived==1]))
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  list(training$Parch[training$Survived == 0], training$Parch[training$Survived == 1])
## Bartlett's K-squared = 1.7309, df = 1, p-value = 0.1883
```

El coeficiente de significancia es mayor que 1, por lo que se puede asumir homogeneidad en las varianzas. Esto podría parecer contradictorio con los gráficos obtenidos, sin embargo, no lo es tanto. Se aprecia que en ambos casos la mayoría de la muestra se sitúa en cero. Si bien es cierto que que entre los supervivientes hay una muestra representativa en el 1, también la distribución de los 0 toma valores más altos, por lo que en conjunto queda compensado.

Fare

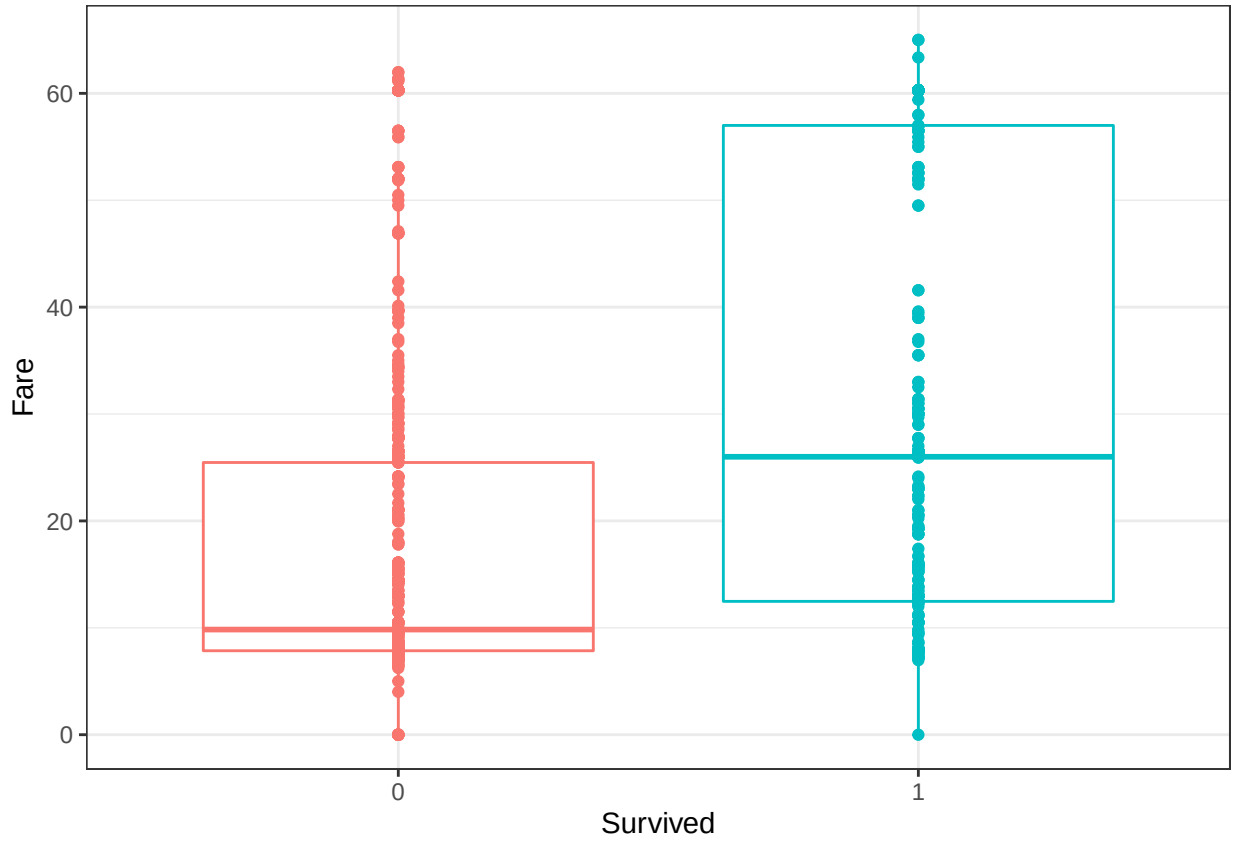
```
# Convert to numeric for the Shapiro test
shapiro.test(training$Fare[training$Survived==0])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$Fare[training$Survived == 0]
## W = 0.75672, p-value < 2.2e-16
```

```r
shapiro.test(training$Fare[training$Survived==1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training$Fare[training$Survived == 1]
## W = 0.83329, p-value < 2.2e-16
```

```r
# Plot data
ggplot(data = training, aes(x = Survived, y = Fare, colour = Survived)) +
  geom_boxplot() +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none")
```




```r
# Validate with Kolmogorov-Smirnov
ks.test(training$Fare[training$Survived==1], pnorm, mean(training$Fare[training$Survived==1]), sd(train
```

```
## Warning in ks.test(training$Fare[training$Survived == 1], pnorm,
## mean(training$Fare[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
```

```
##
## data:  training$Fare[training$Survived == 1]
## D = 0.16346, p-value = 2.314e-08
## alternative hypothesis: two-sided
```

```
ks.test(training$Fare[training$Survived==0], pnorm, mean(training$Fare[training$Survived==0]), sd(train
```

```
## Warning in ks.test(training$Fare[training$Survived == 0], pnorm,
## mean(training$Fare[training$Survived == : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  training$Fare[training$Survived == 0]
## D = 0.22543, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Nuevamente, podemos afirmar que la distribución no es normal, debido al bajísimo p-value obtenido. Si nos fijamos en el gráfico, vemos que en este caso además hay diferencias muy significativas entre el precio del billete de los pasajeros que se salvaron y los que no. Esto daría sentido a la hipótesis de que se salvaron los pasajeros de las clases más altas y que por lo tanto, habían pagado más dinero por su pasaje.

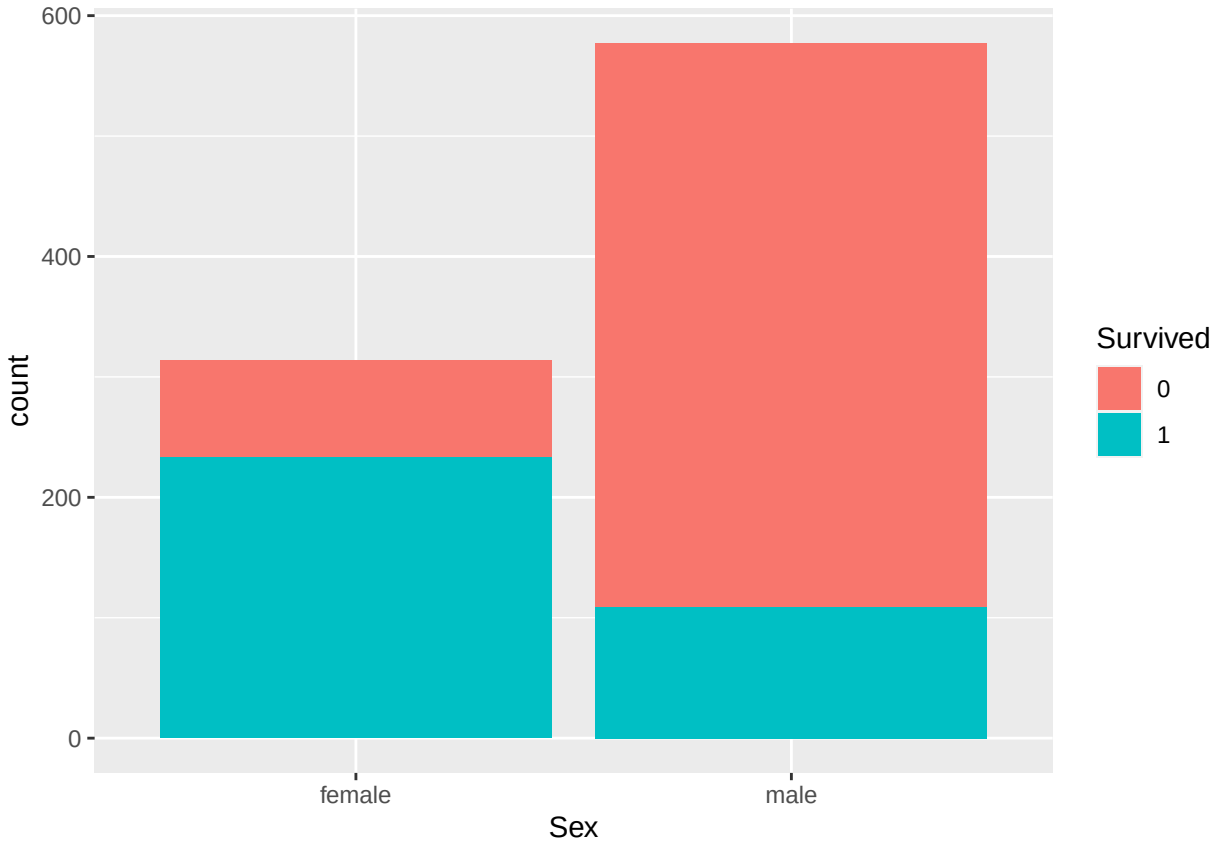
```
bartlett.test(list(training$Fare[training$Survived==0],training$Fare[training$Survived==1]))
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  list(training$Fare[training$Survived == 0], training$Fare[training$Survived == 1])
## Bartlett's K-squared = 48.42, df = 1, p-value = 3.44e-12
```

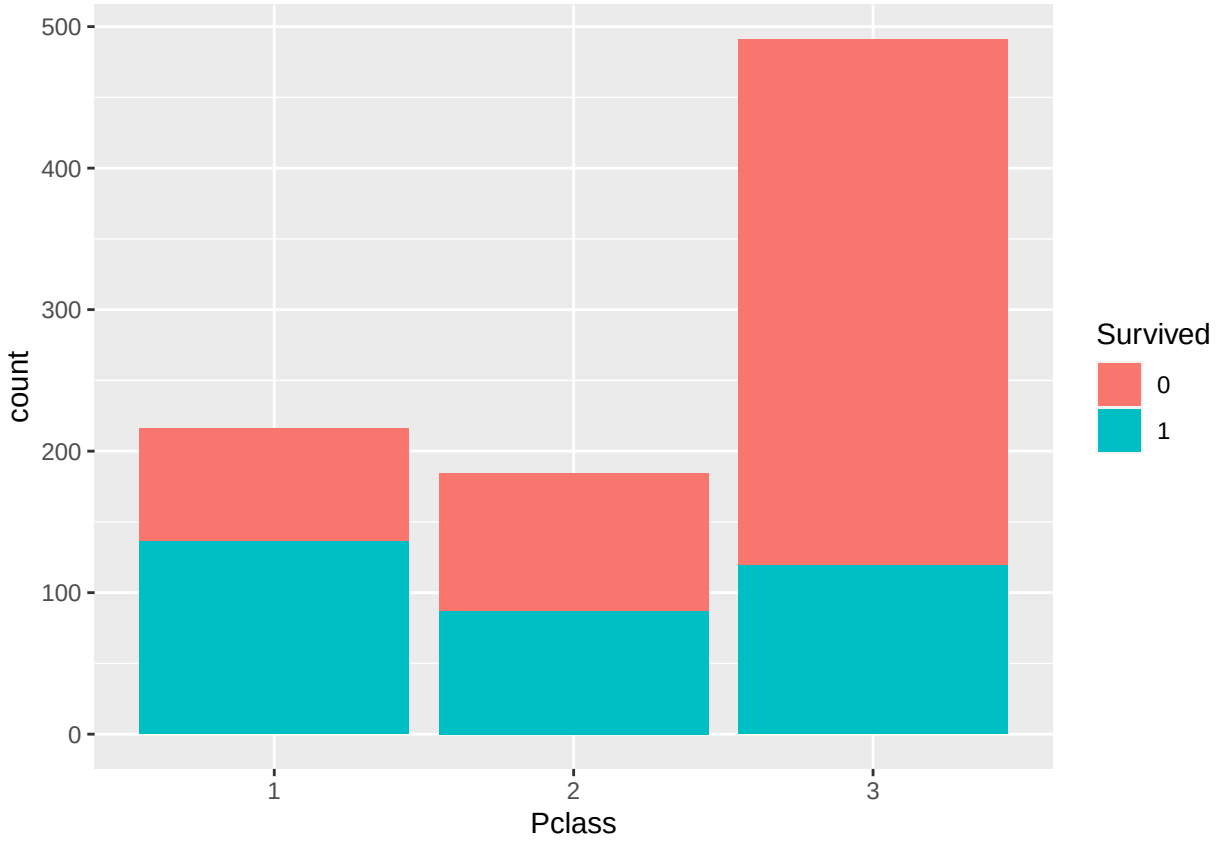
De la aplicación de Barrett, se aprecia que en este caso tampoco se cumple la condición de homogeneidad de la varianza. Todas estas desigualdades entre grupos, tendremos que considerarlas en los siguientes puntos a la hora de hacer comparaciones.

Variables categóricas A continuación, revisamos la representación de las variables catgóricas, que nos servirían para hacer clasificaciones adicionales sobre los grupos que disponemos, creando nuevos grupos de mayor granularidad.

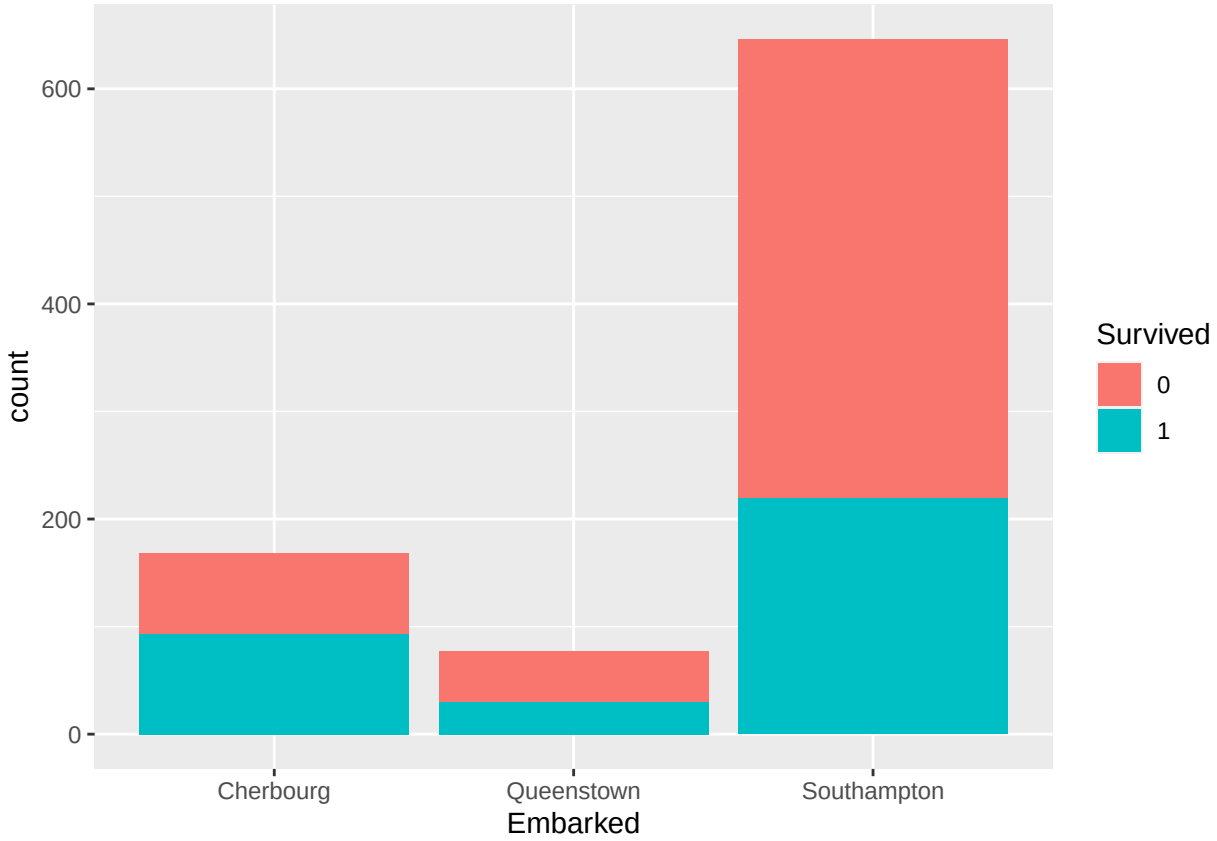
```
library(ggplot2)
library(dplyr)

# Plot sex
ggplot(data=training, aes(x=Sex, fill=Survived))+geom_bar()
```




```
# Plot Pclass
ggplot(data=training, aes(x=Pclass, fill=Survived))+geom_bar()
```




```
# Plot Embarked
ggplot(data=training, aes(x=Embarked, fill=Survived))+geom_bar()
```



Podemos sacar las siguientes conclusiones:

Hay casi el doble de hombres que de mujeres en el dataset.

La mitad de la tripulación del Titanic viajaban en 3 clase.

Casi dos tercios del pasaje embarcaron en Southampton.

Además, se pueden extraer algunas conclusiones de la relación entre las variables categóricas y la variable analizada Survived:

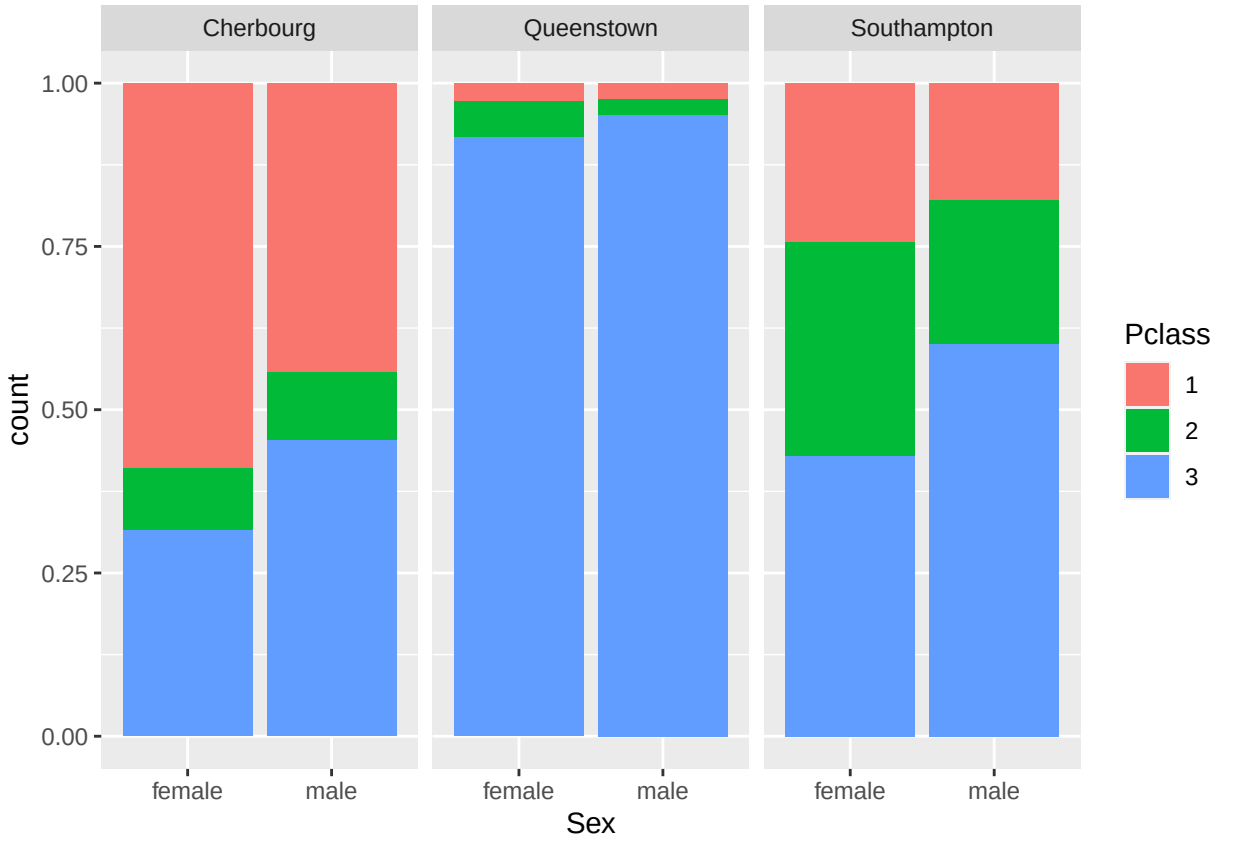
Casi el 70% de las mujeres que iban en el Titanic sobrevivieron, un porcentaje muy superior al de los hombres.

La proporción de supervivientes de pasajeros de las clases 1 y 2 es muy superior a la de 3.

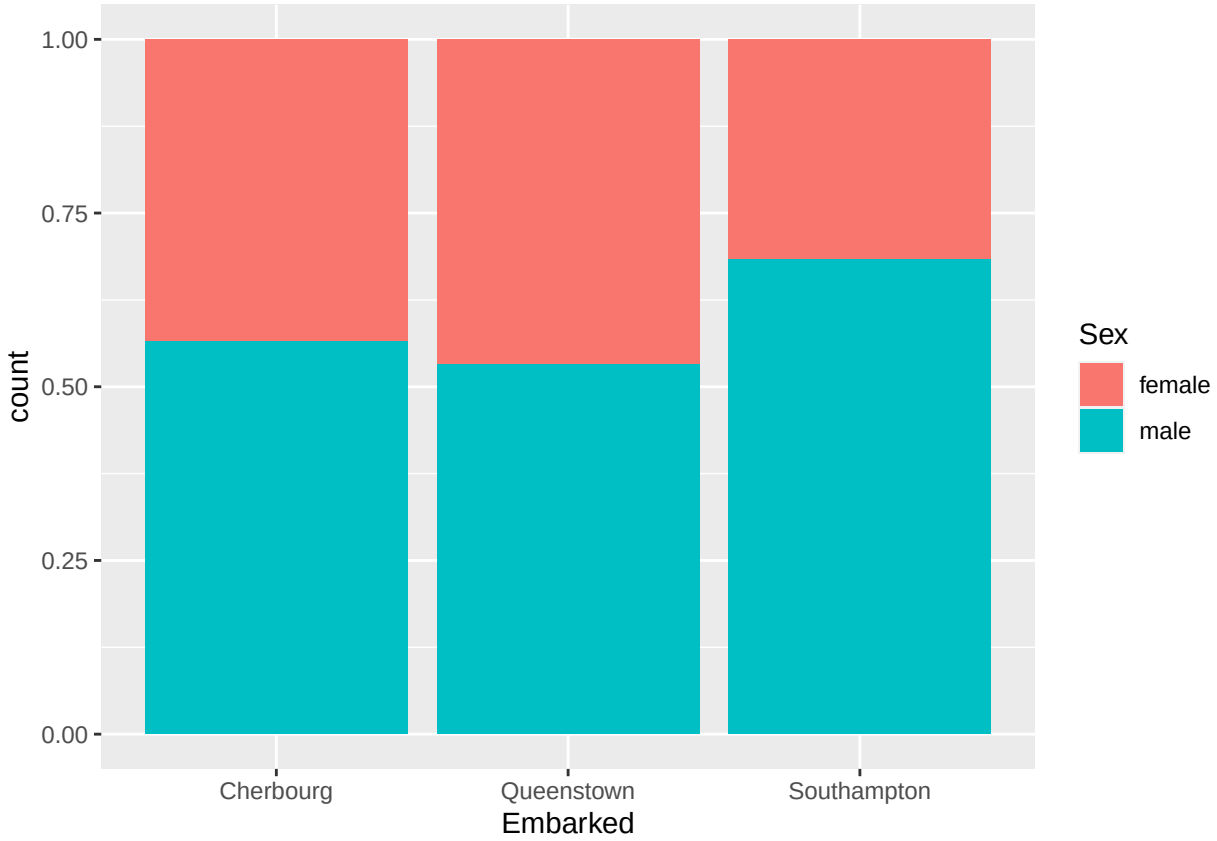
La proporción de supervivientes que embarcaron en Cherbourg es superior a la de Southampton o Queenstown.

Vamos a analizar en detalle el caso de Cherbourg:

```
ggplot(data=training, aes(x=Sex, fill=Pclass))+geom_bar(position = "Fill")+facet_wrap(~Embarked)
```

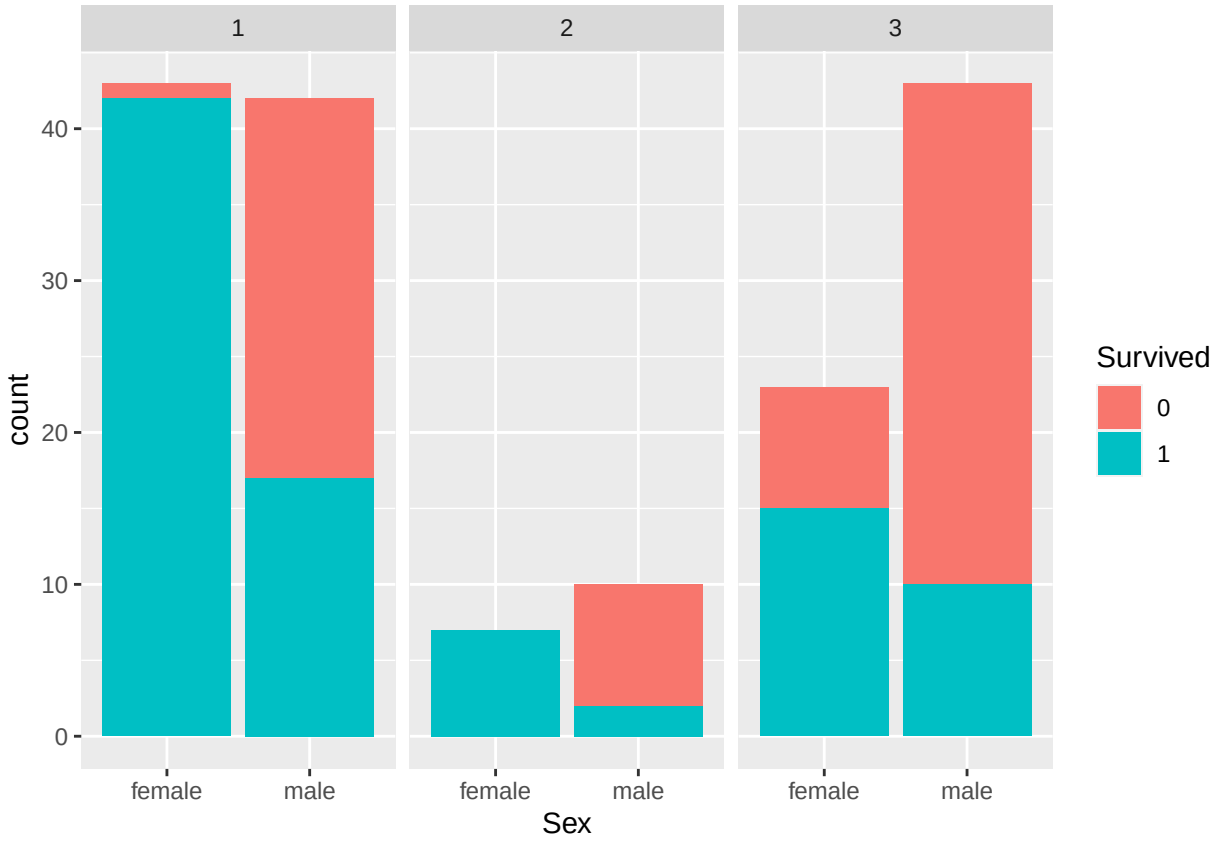



```
ggplot(data=training, aes(x=Embarked, fill=Sex))+geom_bar(position = "Fill")
```



Si analizamos el pasaje por puerto, vemos que en Cherbourg hay un porcentaje superior al 50% de pasajeros de 1ª clase, lo que podría ser un motivo de la alta tasa de supervivencia, además vemos que el porcentaje de mujeres es alto, en torno al 45% de los pasajeros embarcados que es superior al % de mujeres totales en el barco. Además casi el 60% de esas mujeres pertenecían a la clase 1.

```
ggplot(data=filter(training, Embarked=="Cherbourg"), aes(x=Sex, fill=Survived))+geom_bar()+facet_wrap(~
```



###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

Comparación de grupos Al no cumplirse las medidas de homogeneidad de la varianza y la normalidad, no es posible aplicar modelos de contraste de hipótesis de tipo paramétrico, como el método t-Student, sino que tendremos que ir a métodos no paramétricos.

En este apartado, vamos a tratar de comprobar las diferencias que se han podido apreciar del análisis gráfico en los puntos anteriores, entre las variables que parecen más interesantes para el modelo: Cuantitativas:

Fare

Parch

SibSp

Cualitativas:

Sex

Pclass

Descartamos por lo tanto el puerto de embarque, ya que hemos visto que podría tener más relación con la clase y el sexo de los pasajeros, que por el puerto como tal.

Para las variables cuantitativas se aplica el test de Mann-Whitney o el de Wilcoxon, que se aplican ambas de la siguiente forma:

```r
# Fare
wilcox.test(Fare ~ Survived, data = training)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Fare by Survived
## W = 55683, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Parch
wilcox.test(Parch ~ Survived, data = training)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Parch by Survived
## W = 82385, p-value = 3.712e-05
## alternative hypothesis: true location shift is not equal to 0
```

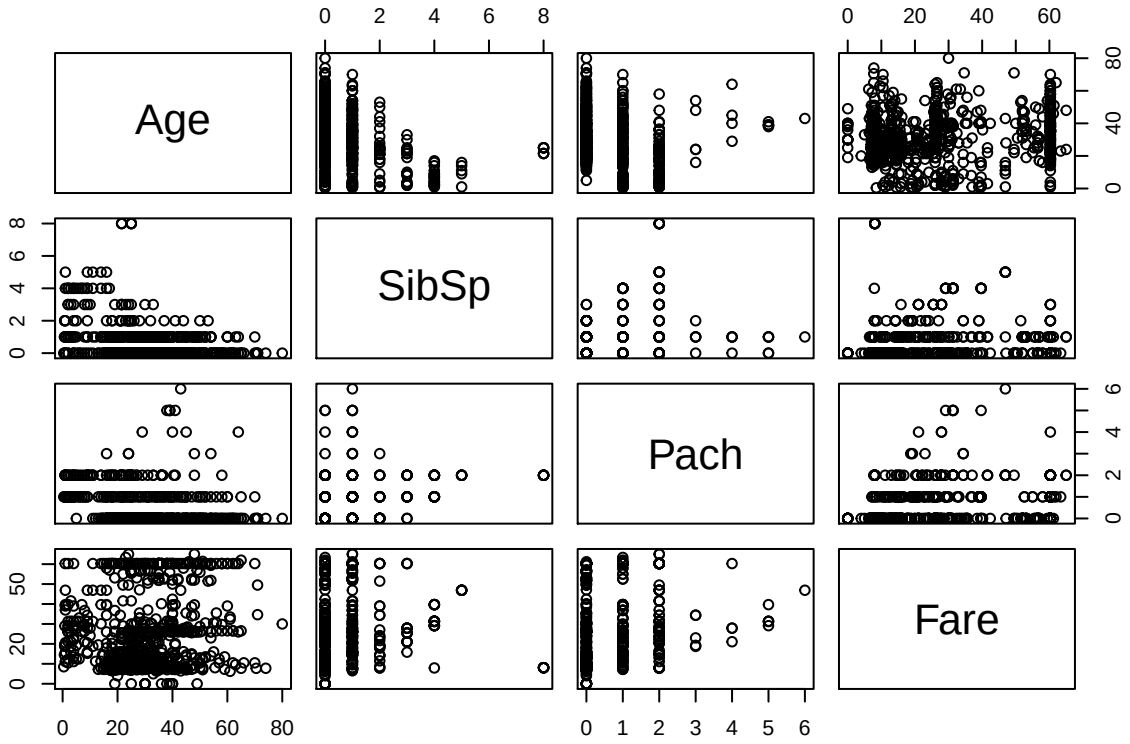
```r
# SibSp
wilcox.test(SibSp ~ Survived, data = training)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  SibSp by Survived
## W = 85775, p-value = 0.008017
## alternative hypothesis: true location shift is not equal to 0
```

Se aprecia que las variables Fare y Parch están, como se presuponía, significativamente diferenciadas en ambos grupos, con valores de significancia muy por debajo del 0.05. En el caso de SibSp, también hay diferencias, sin embargo de un orden mucho menor, por lo que esta variable tendría menos influencia en la determinación de la supervivencia.

Regresión La regresión es un modelo matemático que permite encontrar una relación de dependencia entre una variable dependiente y una o más variables independientes. En este caso, vamos a hacer una prueba para ver la relación que existe entre las variables numéricas:

```r
training_num <- select(training, x=c(Age, SibSp, Parch, Fare))
colnames(training_num) <- c("Age", "SibSp", "Pach", "Fare")
plot(training_num)
```



En los gráficos anteriores podemos ver las cuatro características numéricas que disponemos pintadas una frente a otra. Aunque no se aprecia ningún caso de relaciones lineales, de los gráficos sí que se puede extraer una conclusión: **Conforme aumenta la edad (Age) disminuye el número de SibSp**. Esto tiene sentido porque con 80 años es más complicado que viajen en el barco con 4,5 o 6 hermanas y hermanos o conyuges.

Podemos también utilizar las variables categóricas para crear un modelo de regresión capaz de predecir una variable dicotómica, como es el caso de Survived. Esto lo podemos implementar con un modelo de regresión logística:

```
# Model creation
model_glm <- glm(Survived ~ Pclass+Sex, data=training, family="binomial")
summary(model_glm)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex, family = "binomial", data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1877  -0.7312  -0.4476   0.6465   2.1681
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.2971     0.2190  10.490  < 2e-16 ***
## Pclass2      -0.8380     0.2447  -3.424 0.000618 ***
```

```
## Pclass3      -1.9055     0.2141  -8.898  < 2e-16 ***
## Sexmale      -2.6419     0.1841 -14.351  < 2e-16 ***
## ---
## 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:  826.89  on 887  degrees of freedom
## AIC: 834.89
##
## Number of Fisher Scoring iterations: 4
```

```
# Trusted intervals
confint(model_glm)
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %     97.5 %
## (Intercept)  1.878183  2.7370233
## Pclass2     -1.321995 -0.3615245
## Pclass3     -2.331981 -1.4915248
## Sexmale     -3.010447 -2.2879535
```

Del resumen del modelo vemos que nos da errores relativamente altos, del orden del 20%. Salvo la variable Pclass con valor 2, todas las demás ofrecen valores de significancia muy bajos. Incluso, Pclass2, ofrece un valor muy por debajo del 0.05. Lo que en este caso nos indica que el nivel de significancia está por encima del 95% y por lo tanto las variables del modelo de regresión serían buenas predictoras de la supervivencia con un error en torno al 20%.

Correlación Podemos buscar correlación entre las variables numéricas del dataset. Para ello dado que no se cumplen los criterios de homogeneidad de la varianza y normalidad, utilizaremos el coeficiente de Spearman, que responde mejor a este tipo de poblaciones.

A continuación, buscamos correlaciones entre todos los pares de variables numéricas, utilizando el método de R cor.test() y especificando el método de Spearman:

```
cor.test(training$Age,training$SibSp, method="spearman")
```

```
## Warning in cor.test.default(training$Age, training$SibSp, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  training$Age and training$SibSp
## S = 137482216, p-value = 6.096e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.166179
```

```r
cor.test(training$Age,training$Parch, method="spearman")
```

```
## Warning in cor.test.default(training$Age, training$Parch, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  training$Age and training$Parch
## S = 145651575, p-value = 1.085e-12
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.2354747
```

```r
cor.test(training$Age,training$Fare, method="spearman")
```

```
## Warning in cor.test.default(training$Age, training$Fare, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  training$Age and training$Fare
## S = 95346450, p-value = 8.746e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1912334
```

```r
cor.test(training$Parch,training$SibSp, method="spearman")
```

```
## Warning in cor.test.default(training$Parch, training$SibSp, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  training$Parch and training$SibSp
## S = 64838502, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.450014
```

```r
cor.test(training$Parch,training$Fare, method="spearman")
```

```
## Warning in cor.test.default(training$Parch, training$Fare, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  training$Parch and training$Fare
## S = 72367937, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3861463
```

```
cor.test(training$SibSp,training$Fare, method="spearman")
```

```
## Warning in cor.test.default(training$SibSp, training$Fare, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  training$SibSp and training$Fare
## S = 68977537, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.414905
```

El test de correlación devuelve un resultado entre [-1, 1] donde los extremos indican una alta correlación y el 0 una correlación nula. En el cruce entre las variables anteirores, vemos que en todos los casos nos da valores muy cercanos al cero, por lo que **no existe ninguna correlación entre estas variables**.

Aprendizaje Supervisado Hasta este momento, se han obtenido algunas conclusiones propias del análisis de los datos siguiendo una metodología estadística y tratando de encontrar hipótesis.

Otra alternativa para resolver el ejercicio sería aplicar aprendizaje supervisado, es decir, dejar que un modelo de minería de datos analice el dataset y encuentre una serie de reglas, que podemos comparar con las reglas estadísticas que hemos hayado, para la predicción de la supervivencia.

En este caso, aplicaríamos aprendizaje supervisado porque ya disponemos de un dataset con los dos grupos de interés clasificados, si sobrevive o no, por lo que nos interesa que un modelo sea capaz de predecir la pertenencia a uno de estos grupos. Para ello, utilizaremos un árbol de decisión:

Para la creación del árbol, nos vamos a quedar solamente con los siguientes atributos: Pclass, Sex, Age, Parch y Sib. Como necesitamos que las variables sean categóricas, vamos a hacer una discretización previa de las variables numéricas:

```
# Discretize age
training['ageRange'] <- NaN
training$ageRange[training$Age<=16] <- "Nino"
training$ageRange[training$Age>16] <- "Adulto"
training['ageRange'] <- as.factor(training$ageRange)

# Discretize Parch
training['parchRange'] <- NaN
training$parchRange[training$Parch==0] <- 0
```

```
training$parchRange[training$Parch>0] <- 1
training['parchRange'] <- as.factor(training$parchRange)

# Discretize SibSp
training['sibSpRange'] <- NaN
training$sibSpRange[training$SibSp==0] <- 0
training$sibSpRange[training$SibSp>0] <- 1
training['sibSpRange'] <- as.factor(training$sibSpRange)
```

Una vez hemos preparado los datos, procedemos a crear los datasets de training y test y a crear el modelo:

```
training_dt <- select(training, x=c(Survived, Pclass, Sex, ageRange, parchRange, sibSpRange))
colnames(training_dt) <- c("Survived", "Pclass", "Sex", "ageRange", "parchRange", "sibSpRange")

# Divide between test and training
train_dt <- training_dt[1:600,]
test_dt <- training_dt[600:891,]

# Divide training and test
train_x <- train_dt[,2:6]
train_y <- train_dt[,1]
test_x <- test_dt[,2:6]
test_y <- test_dt[,1]

# Build model
model <- C50::C5.0(train_x, train_y,rules=TRUE )
summary(model)
```

```
##
## Call:
## C5.0.default(x = train_x, y = train_y, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Jan  5 20:27:43 2021
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 600 cases (6 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (226/32, lift 1.4)
##  Pclass = 3
##  Sex = male
##  ->  class 0  [0.855]
##
## Rule 2: (345/55, lift 1.4)
##  Sex = male
##  ageRange = Adulto
##  ->  class 0  [0.839]
##
## Rule 3: (19/1, lift 2.3)
```

```
##  Pclass in {1, 2}
##  ageRange = Nino
##  ->  class 1  [0.905]
##
## Rule 4: (223/56, lift 1.9)
##  Sex = female
##  ->  class 1  [0.747]
##
## Default class: 0
##
##
## Evaluation on training data (600 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       4  117(19.5%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     308    56    (a): class 0
##      61   175    (b): class 1
##
##
##  Attribute usage:
##
##   98.67% Sex
##   60.67% ageRange
##   40.83% Pclass
##
##
## Time: 0.0 secs
```

El árbol es capaz de crear 4 reglas para hacer la clasificación, principalmente utilizando los atributos Sexo, Rango de edad y la clase. Las reglas son las que podemos ver en el resultado del modelo.

Con estas reglas, probamos a clasificar el dataset de test que hemos guardado. Obtenemos un éxito del 78% de precisión.

```
predicted_model <- predict( model, test_x, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_model == test_y) / length(predicte
```

```
## [1] "La precisión del árbol es: 78.0822 %"
```

Representación de los resultados a partir de tablas y gráficas.

Todos los resultados obtenidos se han ido presentando en cada uno de los apartados como conclusión parcial.

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

En este análisis se ha analizado el dataset de pasajeros del Titanic con el objetivo de encontrar variables predictoras de la supervivencia. El problema se ha abordado desde dos puntos de vista: el problema estadístico, con análisis de regresión, correlación y contrastes de hipótesis, y el de la minería de datos, construyendo un árbol de decisión. Vamos a ver las conclusiones en cada caso:

Análisis estadístico El análisis hecho con herramientas estadísticas nos ha arrojado las siguientes conclusiones:

Los grupos de pasajeros que sobrevivieron y que perecieron, **no tienen una distribución normal ni homogeneidad de la varianza**, salvo en el caso de **Parch**.

Casi el **70% de las mujeres** que viajaban en el barco **sobrevivieron**, lo que es casi el doble que el porcentaje de hombres.

El **porcentaje de supervivencia es mucho más alto en pasajeros de primera y segunda clase**, que en los de tercera. Esta afirmación se confirma con el análisis de Fare, que la distribución del precio del billete es más elevada en el caso de los supervivientes. Igual conclusión se obtiene del análisis por puerto de embarque donde el puerto con mayor tasa de supervivencia está directamente relacionado con la clase de los pasajeros que subieron en ese puerto.

El análisis de regresión determina la **clase** y el **género** como las variables más indicadas para predecir el análisis de la supervivencia.

No hemos encontrado **correlación** entre las variables numéricas del dataset.

Del análisis supervisado de minería de datos se han obtenido cuatro reglas para predecir la supervivencia con casi un 80% de precisión:

Los hombres de 3 clase, mueren con una probabilidad del 85%.

Un adulto de 3 clase muere con una probabilidad del 83%.

Un niño de 1 o 2 clase sobrevive con una probabilidad del 90%.

Una mujer de 1 clase sobrevive con una probabilidad del 75%.

Podemos hacer un contraste entre las reglas obtenidas por los métodos anteriores y la famosa frase que ha pasado a la historia de *las mujeres y los niños primero*. Con este análisis, se puede demostrar que el objetivo de garantizar la supervivencia de mujeres y niños se cumplió sólo en parte, dado que del análisis se concluye que definitivamente la clase fue una variable importante también.

Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

Todo el código utilizado para completar la práctica está incluído en este documento.