

TIPOLOGÍA Y CICLO DE VIDA DE LOS DATOS

PRÁCTICA 2

DAVID ARBONA NAVARRO

1. Descripción del dataset.

¿Por qué es importante y qué pregunta/problema pretende responder?

El dataset que se va a utilizar para la práctica proviene de Kaggle es libre y está disponible <https://www.kaggle.com/c/titanic>, nos proporciona información variada de los pasajeros del Titanic, además de la información de si sobrevivieron o no al accidente.

Queremos aprovechar esta información para conocer más detenidamente las características más relevantes que tuvieron las personas que sobrevivieron y las que no lo hicieron.

Las preguntas que queremos resolver son: ¿Qué característica tenían las personas que se salvaron del hundimiento? ¿Tenían alguna característica en común que aumentara la probabilidad de salvarse? ¿Existe relación entre la edad y el tipo de billete que obtuvieron?

Empezaremos echando un primer vistazo al dataset, consta de 891 registros, con 12 atributos que son: *PassengerId*, *Survived*, *Pclass*, *Name*, *Sex*, *Age*, *SibSp*, *Parch*, *Ticket*, *Fare*, *Cabin*, *Embarked*. Hay datos categóricos (*Sex*, *Embarked*), otros discretos como por ejemplo la edad y otros continuos como *Fare*.

```
> 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 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. (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" ...
```

El significado de los atributos es el siguiente:

PassengerId: identificador único del pasajero.

Survived: Sobreviviente 0-NO, 1-SI.

Pclass: Categoría de la clase en la que viajaba 1-Primera, 2-Segunda, 3-Tercera.

Name: Nombre del pasajero.

Sex: Sexo del pasajero Hombre o Mujer.

Age: Edad del pasajero.

SibSp: familiares a bordo.

Parch: padres o hijos que están a bordo.

Ticket: Número del billete.

Fare: Coste del billete.

Cabin: Número de la cabina.

Embarked: Puerto de embarque C = Cherbourg, Q = Queenstown, S = Southampton.

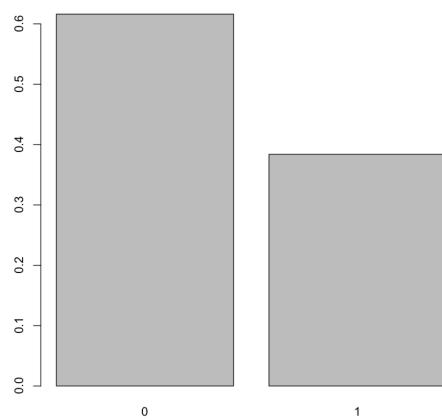
2. Integración y selección de los datos de interés a analizar. (0.5 puntos)

Para el estudio de los datos disponemos de 2 ficheros, uno de ellos (*train*) con todos los campos mencionados y otro (*test*) con todos menos el campo *Survived*, este campo es importante para nuestro análisis y el objetivo de este, por lo que vamos a optar por trabajar únicamente con el conjunto de datos *train* y desechar el conjunto de datos *test*.

Antes de seleccionar los datos de interés para nuestro análisis posterior, realizaremos unas observaciones de los atributos del conjunto de datos para poder tener una visión global sobre ellos:

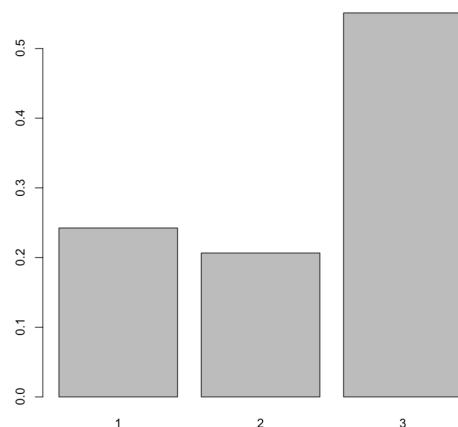
```
# Atributo Survived
prop.table(table(train$Survived))
barplot(prop.table(table(train$Survived)), main="Survived")
```

Survived

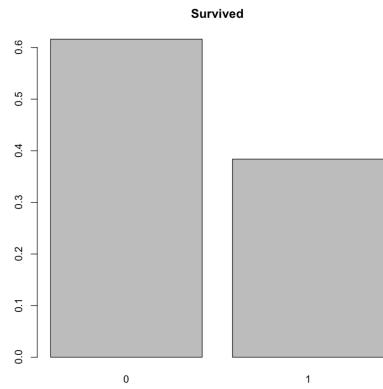


```
# Atributo Pclass
prop.table(table(train$Pclass))
barplot(prop.table(table(train$Pclass)), main="Pclass")
```

Pclass

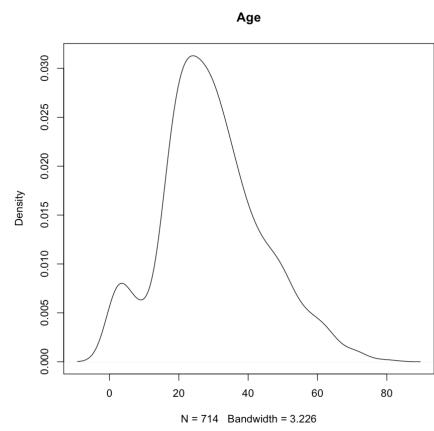


```
# Atributo Sex
prop.table(table(train$Sex))
barplot(prop.table(table(train$Sex)), main="Sex")
```



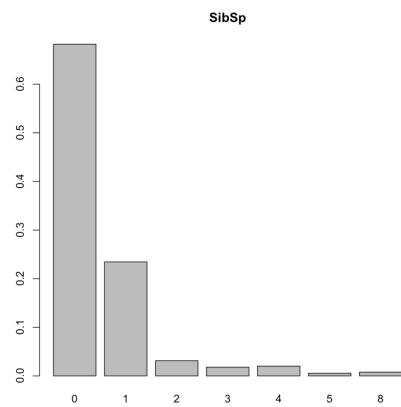
Atributo Age

```
plot(density(na.omit(train$Age)), main = "Age")
```



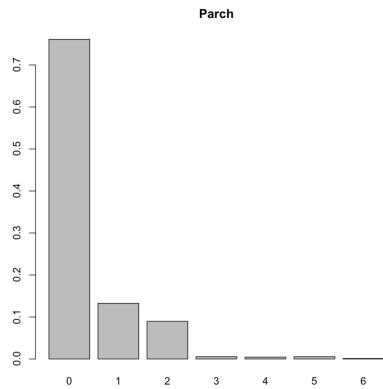
Atributo SibSp

```
prop.table(table(train$SibSp))  
barplot(prop.table(table(train$SibSp)), main="SibSp")
```



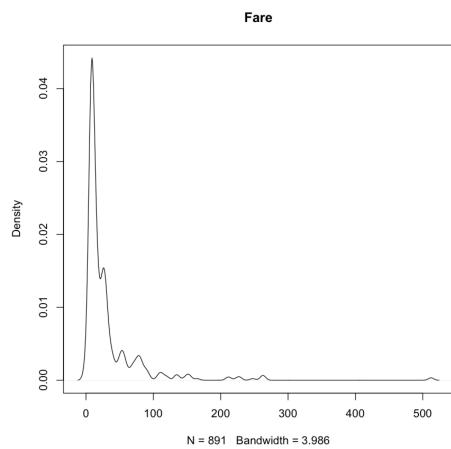
Atributo Parch

```
prop.table(table(train $Parch))  
barplot(prop.table(table(train $Parch)), main="Parch")
```



Atributo Fare

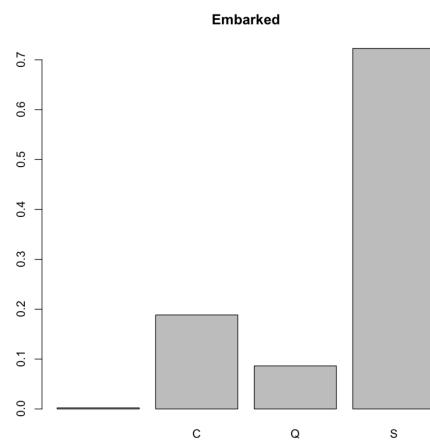
plot(density(na.omit(train\$Fare)),main = "Fare")



Atributo Embarked

prop.table(table(train \$Embarked))

barplot(prop.table(table(train \$Embarked)), main="Embarked")



Observamos los primeros registros de los datos de la siguiente manera:

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

Mediante la instrucción *summary* podemos obtener rápidamente los valores de cada atributo de los datos: mínimo, mediana, media, máximo, ...

```
> summary(train)
  PassengerId   Survived   Pclass      Name       Sex     Age    SibSp    Parch      Ticket      Fare
Min. : 1.0   Min. :0.0000  Min. :1.000  Length:891  Length:891  Min. : 0.42  Min. :0.000  Min. :0.0000  Length:891  Min. : 0.00
1st Qu.:223.5 1st Qu.:0.0000  1st Qu.:2.000  Class :character  Class :character  1st Qu.:20.12  1st Qu.:0.000  1st Qu.:0.0000  Class :character  1st Qu.: 7.91
Median :446.0  Median :0.0000  Median :3.000  Mode  :character  Mode  :character  Median :28.00  Median :0.000  Median :0.0000  Mode  :character  Median :14.45
Mean   :446.0  Mean   :0.3838  Mean   :2.309  NA's   :177       Mean   :29.70  Mean   :0.523  Mean   :0.3816  Mean   :32.20
3rd Qu.:668.5 3rd Qu.:1.0000  3rd Qu.:3.000  NA's   :177       3rd Qu.:38.00  3rd Qu.:1.000  3rd Qu.:0.0000  3rd Qu.:31.00
Max.  :891.0   Max.  :1.0000  Max.  :3.000  NA's   :177       Max.  :80.00  Max.  :8.000  Max.  :6.0000  Max.  :512.33
  Cabin      Embarked
Length:891  Length:891
Class :character  Class :character
Mode  :character  Mode  :character
```

Utilizo la instrucción *str* para saber el número de datos observados, el tipo de datos de cada atributo y un ejemplo de los 10 primeros valores:

```
> 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 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. (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 3 0 1 ...
 $ Parch       : int  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" ...
```

Para el análisis que queremos realizar de los datos nos quedaremos únicamente con los datos que nos aportan valor: *Survived*, *Pclass*, *Sex*, *Age*, *Fare*, *SibSp* y *Parch* de la siguiente manera:

```
train_DC<-train[,c(2, 3, 5, 6, 7, 8, 10)]
```

```
> summary(train_DC)
  Survived   Pclass      Sex     Age    SibSp    Parch      Fare
Min. :0.0000  Min. :1.000  Length:891  Min. : 0.42  Min. :0.000  Min. :0.0000  Min. : 0.00
1st Qu.:0.0000  1st Qu.:2.000  Class :character  1st Qu.:20.12  1st Qu.:0.000  1st Qu.:0.0000  1st Qu.: 7.91
Median :0.0000  Median :3.000  Mode  :character  Median :28.00  Median :0.000  Median :0.0000  Median :14.45
Mean   :0.3838  Mean   :2.309  NA's   :177       Mean   :29.70  Mean   :0.523  Mean   :0.3816  Mean   :32.20
3rd Qu.:1.0000  3rd Qu.:3.000  NA's   :177       3rd Qu.:38.00  3rd Qu.:1.000  3rd Qu.:0.0000  3rd Qu.:31.00
Max.  :1.0000  Max.  :3.000  NA's   :177       Max.  :80.00  Max.  :8.000  Max.  :6.0000  Max.  :512.33
  :
```

3. Limpieza de los datos (2 puntos).

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

Revisando los datos y con las instrucciones de análisis anteriores y con la siguiente (`sapply`), vemos que el único campo que tiene valores vacíos es el `Age`. Es un campo que sí es importante para el propósito de la práctica por lo que los valores los vamos a llenar con la media de todos los valores existentes del campo de esta manera:

```

> sapply(train_DC, function(x) sum(is.na(x)))
Survived   Pclass      Sex      Age     SibSp     Parch     Fare
          0         0         0       177       0         0         0
> train_DC$Age<-ifelse(is.na(train_DC$Age),mean(na.omit(train_DC$Age)), train_DC$Age)
> sapply(train_DC, function(x) sum(is.na(x)))
Survived   Pclass      Sex      Age     SibSp     Parch     Fare
          0         0         0         0       0         0         0

```

Podemos ver que tras la ejecución ya no tenemos valores vacíos.

`Fare` también tienen valores ceros como podemos ver a continuación, en el caso de `Fare` hay muchos pasajeros con valores a 0, sería extraño que un viajero de la clase 1^a viajara gratuitamente, esto nos podría generar datos erróneos a la hora de interpretar los datos. Por otro lado, es correcto que `Survived`, `SibSP` y `Parch` tengan valores 0 que como se ha explicado en el primer punto de la práctica quiere decir que no sobrevivió, o no tuvieron hijos o parentesco con otros pasajeros. También con la segunda ejecución revisamos que no hayan valores de tipo carácter vacíos ("")

```

> sapply(train_DC, function(x) sum(x==0))
Survived   Pclass      Sex      Age     SibSp     Parch     Fare
      549       0         0       0     608     678      15
> sapply(train_DC, function(x) sum(x==""))
Survived   Pclass      Sex      Age     SibSp     Parch     Fare
          0         0         0       0       0         0         0

```

Procedemos a solucionar el problema con el campo `Fare`, analizamos los valores con el comando `summary`, si lo lanzamos para la clase 1^a y para el conjunto entero obtenemos estos datos:

```

>
> summary(train_DC[train_DC$Pclass==1,]$Fare)
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  0.00  30.92  60.29  84.15  93.50  512.33
> summary(train_DC$Fare)
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  0.00  7.91  14.45  32.20  31.00  512.33

```

Para solucionar el problema vamos a realizar el mismo procedimiento que con el campo `Age`, los rellenaremos con la media, pero con la de cada una de las clases económica del billete, ya que si tomáramos la media de todos los pasajeros de todas las clases no sería equitativo y coherente, ya que los de primera clase podrían tener valores muy bajos y los de tercera muy altos y no

reales. Lo realizaremos dividiendo los datos en 3 subconjunto de datos dependiendo de la clase:

```
train_1stClass <- train_DC[train_DC $Pclass == 1,]
train_2stClass <- train_DC[train_DC $Pclass == 2,]
train_3stClass <- train_DC[train_DC $Pclass == 3,]
```

Los valores que son 0 los pasaremos a *NA* para poder realizar la media sin tenerlos en cuenta y sustituirlos con esos valores:

```
train_1stClass$Fare[train_1stClass$Fare == 0] <- NA
train_2stClass$Fare[train_2stClass$Fare == 0] <- NA
train_3stClass$Fare[train_3stClass$Fare == 0] <- NA
```

Modificamos los valores *NA* por la media de su grupo:

```
train_1stClass$Fare<-ifelse(is.na(train_1stClass$Fare),mean(na.omit(train_1stClass$Fare)),
train_1stClass $Fare)
train_2stClass$Fare<-ifelse(is.na(train_2stClass$Fare),mean(na.omit(train_2stClass$Fare)),
train_2stClass$Fare)
train_3stClass$Fare<-ifelse(is.na(train_3stClass$Fare),mean(na.omit(train_3stClass$Fare)),
train_3stClass$Fare)
```

```
> summary (train_1stClass$Fare)
   Min. 1st Qu. Median     Mean 3rd Qu.    Max.
 5.00   33.89  66.60   86.15  93.50  512.33
```

La media es de 86,15, si lo comparamos con la media con los datos sin eliminar los valores 0 que hemos observado anteriormente:

```
>
> summary(train_DC[train_DC$Pclass==1,]$Fare)
   Min. 1st Qu. Median     Mean 3rd Qu.    Max.
 0.00   30.92  60.29   84.15  93.50  512.33
```

Vemos que la media es menor en este caso, afectados por estos datos y pudiendo distorsionar nuestros análisis.

Finalmente unimos los 3 subconjuntos para volver a obtener el conjunto de datos ya con los datos del campo *Fare* correctos:

```
> train_DC<-rbind(train_1stClass, train_2stClass, train_3stClass)
> summary(train_DC$Fare)
   Min. 1st Qu. Median     Mean 3rd Qu.    Max.
 4.013   7.925  14.500  32.893  31.275 512.329
> summary(train_DC)
  Survived      Pclass       Sex        Age      SibSp      Parch       Fare
Min.   :0.0000  Min.   :1.000  Length:891  Min.   :0.42  Min.   :0.0000  Min.   : 4.013
1st Qu.:0.0000  1st Qu.:2.000  Class :character  1st Qu.:22.00  1st Qu.:0.0000  1st Qu.: 7.925
Median :0.0000  Median :3.000  Mode  :character  Median :29.70  Median :0.0000  Median :14.500
Mean   :0.3838  Mean   :2.309  NA's   :278    Mean   :29.70  Mean   :0.523   Mean   :32.893
3rd Qu.:1.0000  3rd Qu.:3.000           NA's   :145    3rd Qu.:35.00  3rd Qu.:1.000   3rd Qu.:31.275
Max.   :1.0000  Max.   :3.000           NA's   :22     Max.   :80.00  Max.   :8.000   Max.   :512.329
```

2. Identificación y tratamiento de valores extremos.

Para revisar los valores extremos tenemos que observar aquellos atributos que no son categóricos, en nuestro caso *Age* y *Fare*, en el primero de ellos vemos valores extremos como el valor de 80, y por abajo cercano, pero en este caso tienen sentido y son correctos ya que representan la edad de las personas y son valores totalmente reales y coherentes, por lo que no vamos a realizar ningún tipo de trasformación ni modificación.

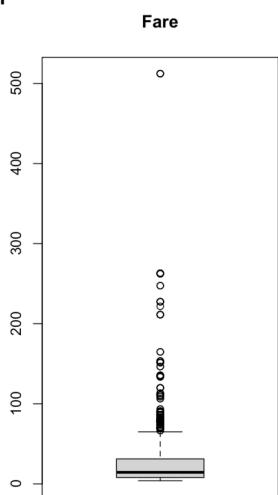


```

> a<-boxplot(train_data_clean$Age,main="Age")
> a$out
[1] 2.00 58.00 55.00 2.00 66.00 65.00 0.83 59.00 71.00 70.50 2.00 55.50 1.00 61.00 1.00 56.00
[17] 1.00 58.00 2.00 59.00 62.00 58.00 63.00 65.00 2.00 0.92 61.00 2.00 60.00 1.00 1.00 64.00
[33] 65.00 56.00 0.75 2.00 63.00 58.00 55.00 71.00 2.00 64.00 62.00 62.00 60.00 61.00 57.00 80.00
[49] 2.00 0.75 56.00 58.00 70.00 60.00 60.00 70.00 0.67 57.00 1.00 0.42 2.00 1.00 62.00 0.83
[65] 74.00 56.00

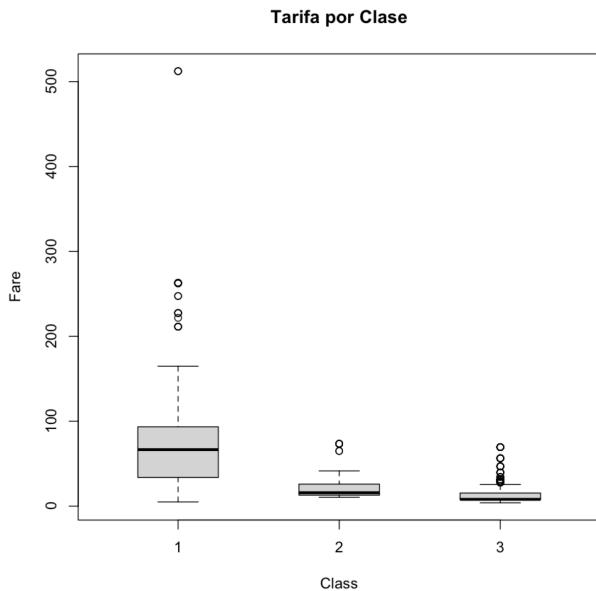
```

En cambio, cuando analizamos los *outliers* del campo *Fare* mediante la función boxplot (gráfica siguiente), podemos apreciar que hay muchos valores elevados, podemos apreciar que hay cierta coherencia en los valores cercanos a 200, entendemos que corresponden a la 1^a clase cuyas tarifas eran más altas, pero en cambio vemos un valor que sobresale del resto y es mayor a 500 y no parece coherente.



```
> a<-boxplot(train_data_clean$Fare,main="Fare")
> a$out
[1] 71.2833 263.0000 146.5208 82.1708 76.7292 80.0000 83.4750 73.5000 263.0000 77.2875
[11] 247.5208 73.5000 77.2875 79.2000 66.6000 69.5500 69.5500 146.5208 69.5500 113.2750
[21] 76.2917 90.0000 83.4750 90.0000 79.2000 86.5000 512.3292 79.6500 153.4625 135.6333
[31] 77.9583 78.8500 91.0792 151.5500 247.5208 151.5500 110.8833 108.9000 83.1583 262.3750
[41] 164.8667 134.5000 69.5500 135.6333 153.4625 133.6500 66.6000 134.5000 263.0000 75.2500
[51] 69.3000 135.6333 82.1708 211.5000 227.5250 73.5000 120.0000 113.2750 90.0000 120.0000
[61] 263.0000 81.8583 89.1042 91.0792 90.0000 78.2667 151.5500 86.5000 108.9000 93.5000
[71] 221.7792 106.4250 71.0000 106.4250 110.8833 227.5250 79.6500 110.8833 79.6500 79.2000
[81] 78.2667 153.4625 77.9583 69.3000 76.7292 73.5000 113.2750 133.6500 73.5000 512.3292
[91] 76.7292 211.3375 110.8833 227.5250 151.5500 227.5250 211.3375 512.3292 78.8500 262.3750
[101] 71.0000 86.5000 120.0000 77.9583 211.3375 79.2000 69.5500 120.0000 93.5000 80.0000
[111] 83.1583 69.5500 89.1042 164.8667 69.5500 83.1583
```

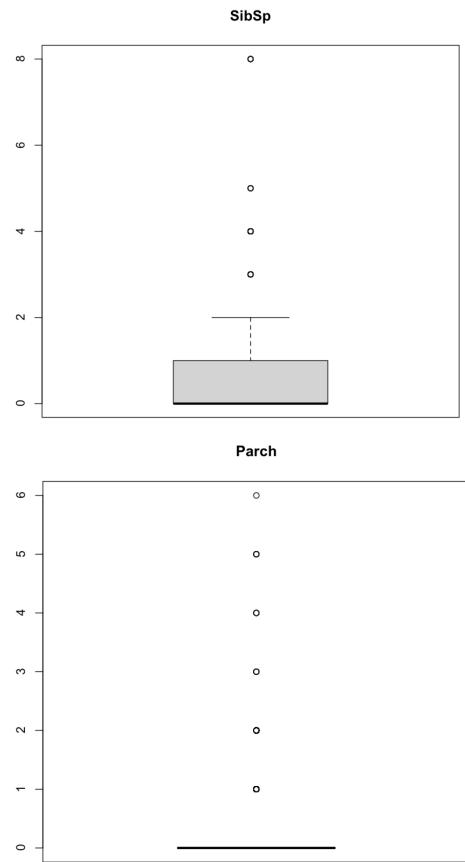
En este caso vamos a examinar los datos relacionándolo con el campo clase como vemos en el siguiente boxplot, en él podemos apreciar que el valor de más de 500 corresponde a la primera clase, por lo que podría ser que fueran los que más pagaran.



Revisando los datos originales (train) con detenimiento de estos outliers, vemos que hay 3 personas y que al parecer revisando los datos sí son coherentes, y es correcto esa tarifa.

```
> train[train$Fare > 500,]
   PassengerId Survived Pclass          Name     Sex Age SibSp Parch Ticket
259           259       1     1    Ward, Miss. Anna female  35    0    0 PC 17755
680           680       1     1 Cardeza, Mr. Thomas Drake Martinez male  36    0    1 PC 17755
738           738       1     1    Lesurer, Mr. Gustave J male  35    0    0 PC 17755
               Fare Cabin Embarked
259 512.3292      C
680 512.3292  B51 B53 B55      C
738 512.3292      B101      C
```

Por lo que no realizaré ningún tipo de tratamiento sobre los valores extremos (*outliers*) ya que al parecer son correctos.



```
> a3<-boxplot(train_DC $SibSp,main="SibSp")
> a3$out
[1] 3 3 3 3 3 4 3 4 5 3 4 5 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 5 3 5 3 4 4 3 3 5 4 4 8 4 3 4 8 4 8
> a4<-boxplot(train_DC $Parch,main="Parch")
> a4$out
[1] 2 1 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 2 1 1 1 2 1 2 2 2 4 2 2 1 2 2 2 1 1 2 1 2 1 1 2 1 2 1 2 1 1 1 1 2 2 2 1 1 2 1 1 2 1 1 2 1 1 5 1 5 1 2 2 1 2 2 3 2 2 1 2 2 2 1 1 2 1 1 2 5 2 1 1 6 2 1 1
[64] 1 1 1 1 1 1 1 2 3 1 1 2 2 2 1 1 2 1 1 1 2 2 1 1 1 1 2 1 2 1 3 1 1 2 1 1 1 1 2 1 1 5 1 5 1 2 2 1 2 2 3 2 2 1 2 2 2 1 1 2 1 1 2 1 1 2 1 1 4 1 1 2 2 2 1 2 1 1 2 1 1 1 2 2 2 1 4 2 1 5 1 1 2 5 2 1 1
[127] 2 4 1 1 1 1 2 2 2 1 1 2 1 1 2 2 2 1 1 1 2 1 1 1 4 1 1 2 2 2 1 2 1 1 2 1 1 1 2 2 2 1 4 2 1 5 1 1 2 5 2 1 1
[190] 1 1 3 1 1 2 1 2 2 1 1 2 2 1 1 2 2 1 1 3 2 1 5 2
```

En el caso de los campos *Parch* y *SibSp* podemos observar valores máximos de 8 y 6 respectivamente, que teniendo en cuenta el significado de estas variables son completamente posibles, por lo que, no realizaremos tampoco ningún control de estos valores extremos.

4. Análisis de los datos (2,5 puntos).

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

Como se ha explicado en el punto 2 de la práctica se ha realizado una selección de los datos originales de interés, para finalmente trabajar con los siguientes:

```
> summary(train_DC)
   Survived         Pclass          Sex            Age           SibSp          Parch          Fare
Min.   :0.0000   Min.   :1.000   Length:891   Min.   :0.42   Min.   :0.000   Min.   :0.0000   Min.   : 4.013
1st Qu.:0.0000  1st Qu.:2.000   Class  :character  1st Qu.:22.00  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.: 7.925
Median :0.0000  Median :3.000   Mode   :character  Median :29.70  Median :0.000   Median :0.0000   Median :14.500
Mean   :0.3838  Mean   :2.309   NA's    :891       Mean   :29.70  Mean   :0.523   Mean   :0.3816   Mean   :32.893
3rd Qu.:1.0000  3rd Qu.:3.000   NA's    :891       3rd Qu.:35.00  3rd Qu.:1.000   3rd Qu.:0.0000   3rd Qu.:31.275
Max.   :1.0000  Max.   :3.000   NA's    :891       Max.   :80.00  Max.   :8.000   Max.   :6.0000   Max.   :512.329
```

Vamos a trabajar con todos ellos para obtener análisis cuando sea posible teniendo en cuenta el tipo de datos que son: discretos, continuos o categóricos.

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

Para comprobar la normalidad de la varianza se utilizará el test de Shapiro-Wilk es uno de los métodos más potentes y el test de Kolmogorov-Smirnov para los datos numéricos *Age* y *Fare*. Los realizaremos de la siguiente manera:

```
>
> shapiro.test(train_DC$Age)

Shapiro-Wilk normality test

data: train_DC$Age
W = 0.95882, p-value = 3.969e-15

> ks.test(train_DC$Age, pnorm, mean(train_DC$Age), sd(train_DC$Age))

One-sample Kolmogorov-Smirnov test

data: train_DC$Age
D = 0.14846, p-value < 2.2e-16
alternative hypothesis: two-sided


> shapiro.test(train_DC$Fare)

Shapiro-Wilk normality test

data: train_DC$Fare
W = 0.52241, p-value < 2.2e-16

> ks.test(train_DC$Fare, pnorm, mean(train_DC$Fare), sd(train_DC$Fare))

One-sample Kolmogorov-Smirnov test

data: train_DC$Fare
D = 0.29389, p-value < 2.2e-16
alternative hypothesis: two-sided
```

Como podemos apreciar en el valor de p-valor el número obtenido es muy bajo, mucho menos que el valor de significancia que generalmente es de 0.05. Por lo que la hipótesis nula se rechaza y por tanto los datos de ninguno de los dos campos cuentan con una distribución normal.

Para comprobar la homogeneidad de la varianza (homocedasticidad) ya que los campos no siguen una distribución normal, utilizaremos el test de Fligner-Killeen, para distintas combinaciones de campos para poder entender mejor los datos, de la siguiente manera:

```
> fligner.test(Age ~ Pclass, data = train_DC)

Fligner-Killeen test of homogeneity of variances

data: Age by Pclass
Fligner-Killeen:med chi-squared = 34.97, df = 2, p-value = 2.55e-08

> fligner.test(Age ~ Survived, data = train_DC)

Fligner-Killeen test of homogeneity of variances

data: Age by Survived
Fligner-Killeen:med chi-squared = 5.4227, df = 1, p-value = 0.01988

> fligner.test(Fare ~ Survived, data = train_DC)

Fligner-Killeen test of homogeneity of variances

data: Fare by Survived
Fligner-Killeen:med chi-squared = 90.15, df = 1, p-value < 2.2e-16

> fligner.test(Fare ~ Pclass, data = train_DC)

Fligner-Killeen test of homogeneity of variances

data: Fare by Pclass
Fligner-Killeen:med chi-squared = 378.99, df = 2, p-value < 2.2e-16

> fligner.test(Age ~ Sex, data = train_DC)

Fligner-Killeen test of homogeneity of variances

data: Age by Sex
Fligner-Killeen:med chi-squared = 1.052, df = 1, p-value = 0.305

> fligner.test(Fare ~ Sex, data = train_DC)

Fligner-Killeen test of homogeneity of variances

data: Fare by Sex
Fligner-Killeen:med chi-squared = 49.914, df = 1, p-value = 1.607e-12
```

En los datos apreciamos que el *p-valor* es inferior al nivel de significancia (0,05) en la mayoría de los casos por lo que indican heterocedasticidad, esto nos informa que la varianza del campo Fare tiene valores diferentes para los distintos grupos de Pclass, Sex y Survived; para el campo Age esto ocurre con los distintos grupos de Pclass y Survived.

Únicamente en el caso de la Edad por Sexos vemos que sí es mayor que el nivel de significancia por lo que en este caso no se rechaza la hipótesis nula de homocedasticidad, que asume que tiene valores iguales de varianza entre los grupos de datos de tipo Sexo (masculino, femenino) para la variable Age.

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.

Primero realizaremos pruebas de contraste de hipótesis para comparar grupos de datos. Debido a que hemos visto que no se cumplen la normalidad y la homocedasticidad tendremos que realizarlos mediante pruebas no paramétricas, como Wilcoxon (que es la que utilizaremos ya que los datos son dependientes) o Mann-Whitney (cuando son independientes). Podemos utilizar este método cuando se vayan a comparar dos grupos de datos, en nuestro caso los grupos serán el Sexo y por otro lado si sobrevivió o no. Los resultados los podemos ver a continuación:

```
> wilcox.test(Age ~ Sex, data = train_DC)

Wilcoxon rank sum test with continuity correction

data: Age by Sex
W = 83196, p-value = 0.04309
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(Age ~ Survived, data = train_DC)

Wilcoxon rank sum test with continuity correction

data: Age by Survived
W = 98220, p-value = 0.2434
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(Fare ~ Sex, data = train_DC)

Wilcoxon rank sum test with continuity correction

data: Fare by Sex
W = 116334, p-value = 2.27e-12
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(Fare ~ Survived, data = train_DC)

Wilcoxon rank sum test with continuity correction

data: Fare by Survived
W = 60160, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

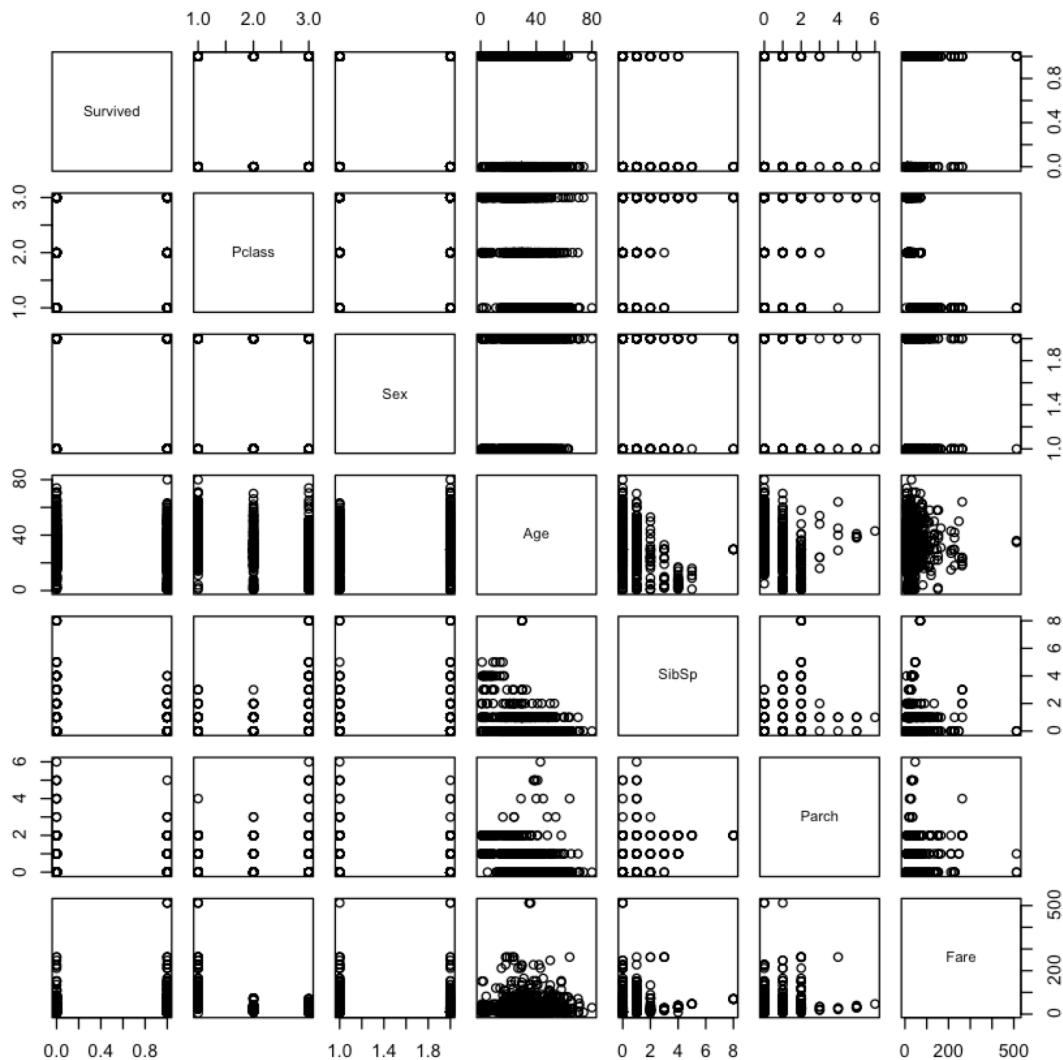
Vemos que el p-valor únicamente entre los grupos de datos Survived para la variable Age, es mayor que el nivel de significancia, lo que nos informa que no existen diferencias estadísticamente significativas en la edad de los que sobrevivieron y los que no.

En el resto de test Wilcoxon el p-valor es inferior al nivel de significancia, por lo que la hipótesis nula es rechazada por lo que existen diferencias significativas entre los grupos de datos analizados.

REGRESIÓN:

Primero que nada, mediante la función plot vamos a examinar los datos en conjunto:

`plot(train_DC)`



Aplicaremos regresión lineal utilizando la función lm de R, la realizaremos múltiple para observar los resultados entre variables de nuestro dataset:

```

> m1 = lm(Age~Pclass,data=train_DC)
> summary(m1)

Call:
lm(formula = Age ~ Pclass, data = train_DC)

Residuals:
    Min      1Q  Median      3Q     Max 
-35.522 -6.743  0.711  4.711 47.863 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 41.5950    1.2082  34.43 <2e-16 ***
Pclass       -5.1528    0.4921 -10.47 <2e-16 ***  
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.27 on 889 degrees of freedom
Multiple R-squared:  0.1098, Adjusted R-squared:  0.1088 
F-statistic: 109.6 on 1 and 889 DF,  p-value: < 2.2e-16

> m3 = lm(Fare~ Pclass,data=train_DC)
> summary(m3)

Call:
lm(formula = Fare ~ Pclass, data = train_DC)

Residuals:
    Min      1Q  Median      3Q     Max 
-71.82 -17.25 -1.83   4.81 435.51 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 110.392    4.040   27.33 <2e-16 ***
Pclass       -33.569   1.645  -20.40 <2e-16 ***  
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41.04 on 889 degrees of freedom
Multiple R-squared:  0.3189, Adjusted R-squared:  0.3181 
F-statistic: 416.2 on 1 and 889 DF,  p-value: < 2.2e-16

```

Podemos observar que de nuestro conjunto de datos el que mayor *R-Squared* (coeficiente de determinación) es la relación de la tarifa (*Fare*) con la clase (*Pclass*), aún así el valor es 0.3189 que dista mucho del valor 1, por lo que la relación no es tan alta como se esperaba.

Si realizamos modelos polinómicos más complejos que permite la función *lm* relacionando la clase con la tarifa de forma cuadrática mejora el *R-Squared* hasta el valor de 0.48.

```

> m4 = lm(Pclass~Fare+I(Fare^2),data=train_DC)
> summary(m4)

Call:
lm(formula = Pclass ~ Fare + I(Fare^2), data = train_DC)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.7624 -0.4534  0.2924  0.3298 1.4167 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.867e+00 2.843e-02 100.85 <2e-16 ***
Fare        -2.115e-02 8.117e-04 -26.05 <2e-16 ***  
I(Fare^2)   3.862e-05 2.328e-06 16.59 <2e-16 ***  
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6036 on 888 degrees of freedom
Multiple R-squared:  0.48,  Adjusted R-squared:  0.4788 
F-statistic: 409.8 on 2 and 888 DF,  p-value: < 2.2e-16

```

CORRELACIONES:

En el caso de estudio, para calcular la correlación entre variables vamos a utilizar el método Spearman, ya que ninguno de los datos sigue una distribución

normal. Al ejecutarlo sobre dos de los campos que entendemos que tiene una mayor correlación obtenemos el siguiente resultado:

```
> cor.test(train_DC$Fare, train_DC$Pclass, method="spearman")
Spearman's rank correlation rho

data: train_DC$Fare and train_DC$Pclass
S = 203947421, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
-0.7299634
```

Se obtiene un valor de -0.7299, es negativo porque como sabemos cuánto mayor es la tarifa (mayor precio) la clase es menor (primera clase), y por tanto la correlación es negativa. Es un valor alto por lo que tienen una alta correlación.

Revisando el resto de correlaciones entre campos del dataset vemos que es muy baja, menor de 0.34 en el mejor de los casos, para la que hay entre los campos de *PClass* y *Survived*, que puede tener cierta lógica a priori, pero que vemos que no es así finalmente.

```
> cor.test(train_DC$Age, train_DC$Pclass, method="spearman")
Spearman's rank correlation rho

data: train_DC$Age and train_DC$Pclass
S = 154304817, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
-0.308875

> cor.test(train_DC$Age, train_DC$SibSp, method="spearman")
Spearman's rank correlation rho

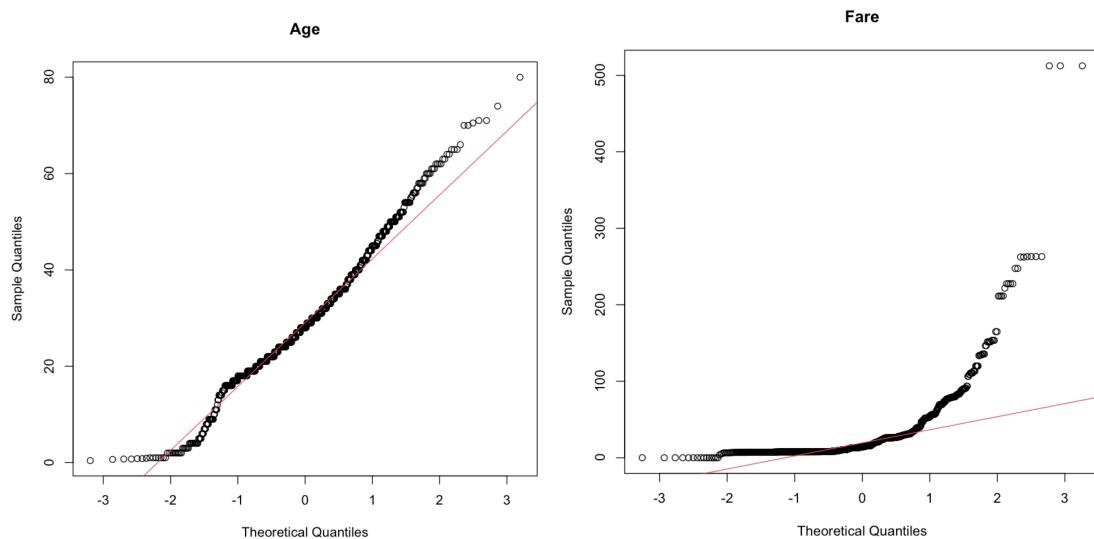
data: train_DC$Age and train_DC$SibSp
S = 135225253, p-value = 1.049e-05
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
-0.1470345

> cor.test(train_DC$Pclass, train_DC$Survived, method="spearman")
| Spearman's rank correlation rho
| 
| data: train_DC$Pclass and train_DC$Survived
| S = 157935034, p-value < 2.2e-16
| alternative hypothesis: true rho is not equal to 0
| sample estimates:
|   rho
| -0.3396679
```

5. Representación de los resultados a partir de tablas y gráficas. (2 puntos)

Para el estudio de los datos es importante realizar representaciones gráficas de los mismos, en el punto 2 de la práctica, para analizar el conjunto de datos y conocerlo mejor, ya hemos obtenido distintas gráficas representativas y donde podemos ver la distribución de los datos y sus valores. Vamos a seguir sacando información de los datos con distintos comandos y gráficas:

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

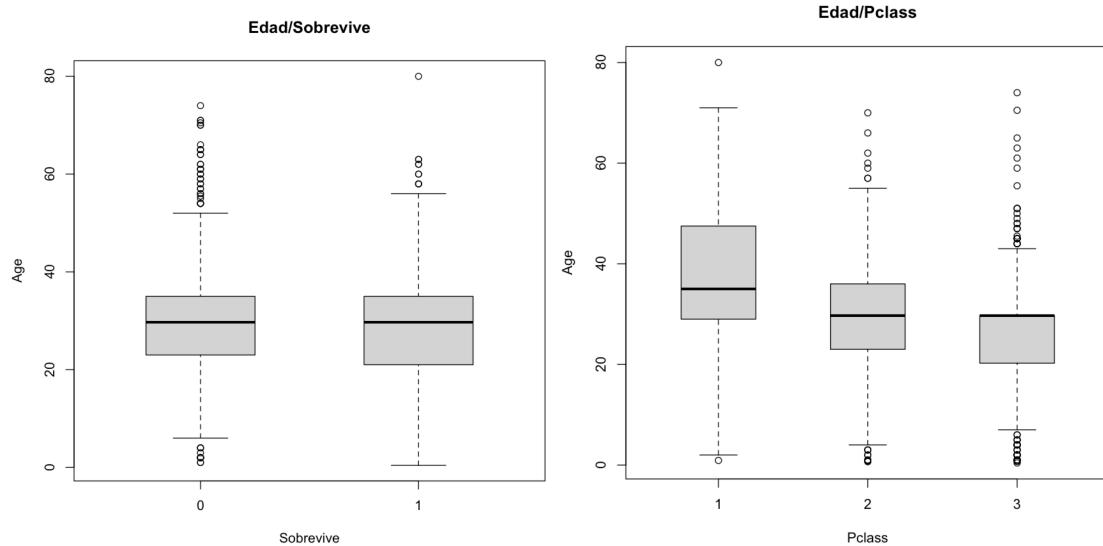


Podemos apreciar en el campo Age que está cercana a la normalidad, mientras que el campo Fare no.

Mediante los bloxpot podemos ver la relación entre conjuntos de datos tal y como también hemos analizado en el punto 3 de la práctica, lo realizaremos de la siguiente manera:

```
boxplot (train_data_clean$Age~train_data_clean$Survived, data=train_data_clean,
xlab="Sobrevive", ylab="Age", main="Edad por Sobrevive",boxwex=0.5)

boxplot (train_data_clean$Age~train_data_clean$Pclass, data=train_data_clean, xlab="Pclass",
ylab="Age", main="Edad por Pclass",boxwex=0.5)
```



Otra manera muy útil de obtener información de los datos es mediante la instrucción barplot agrupándolo por si sobreviven o no, podemos ver cómo lo hemos obtenido por Edad, por Clase, por Sexo, por familiares a bordo y por número de hijos.

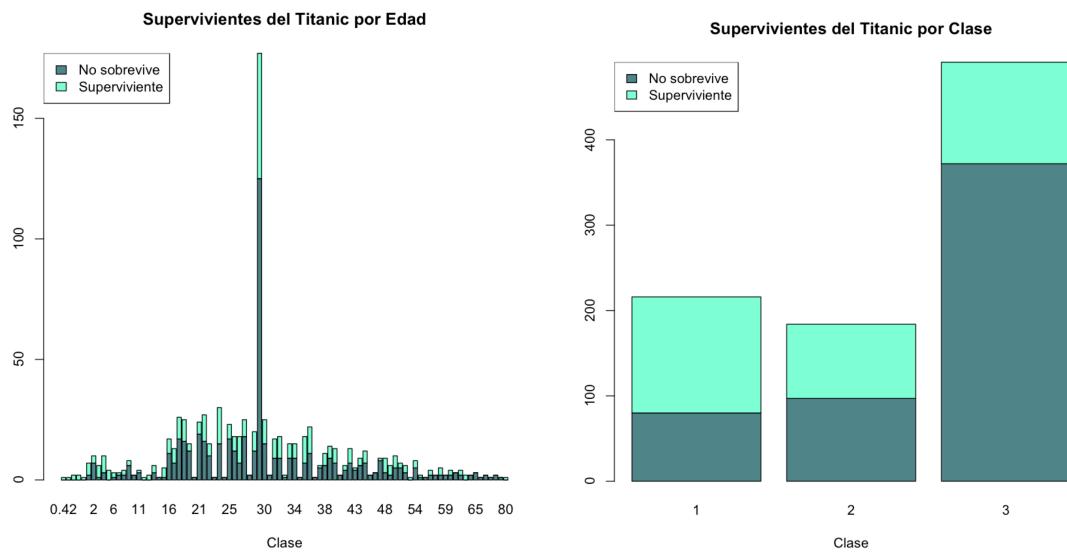
```

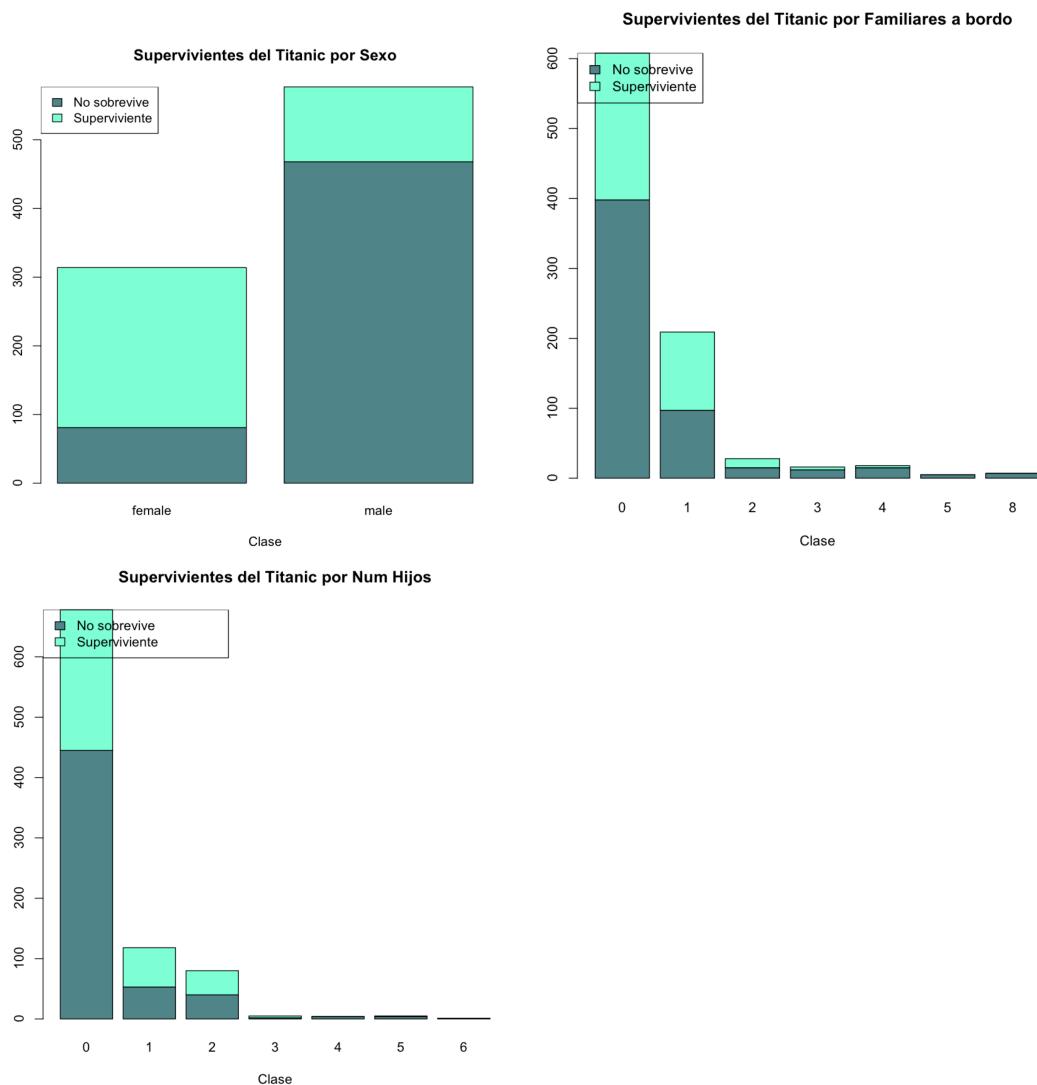
train_AgeSurvived<-table(train_DC[,c(1,4)])
barplot(train_AgeSurvived, main = "Supervivientes del Titanic", xlab = "Clase", col =
c("cadetblue4","aquamarine"))
legend("topleft", c("No sobrevive","Superviviente"), fill = c("cadetblue4","aquamarine"))

train_ClaseSurvived<-table(train_DC[,c(1,2)])
barplot(train_ClaseSurvived, main = "Supervivientes del Titanic por Clase", xlab = "Clase", col =
c("cadetblue4","aquamarine"))
legend("topleft", c("No sobrevive","Superviviente"), fill = c("cadetblue4","aquamarine"))

train_SexoSurvived<-table(train_DC[,c(1,3)])
barplot(train_SexoSurvived, main = "Supervivientes del Titanic por Clase", xlab = "Clase", col =
c("cadetblue4","aquamarine"))
legend("topleft", c("No sobrevive","Superviviente"), fill = c("cadetblue4","aquamarine"))

```





A partir de estas gráficas de barras podemos sacar gran cantidad de información, por ejemplo, se aprecia que más de la mitad de las personas que viajaron en 1^a clase sobrevivieron, mientras que los que viajaron en 3^a sólo sobre el 25%.

Por otro lado, cuando obtenemos la información por sexo, vemos que la gran mayoría que sobrevivieron fueron mujeres sobre el 75%, mientras que de hombre menos del 20% aproximadamente.

6. Resolución del problema. (0.5 puntos)

A partir de los resultados obtenidos, ¿Cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Gracias a la información extraída en el análisis de los datos podemos contestar a las preguntas que nos realizamos inicialmente.

Con las correlaciones entre campos hemos podido apreciar distintas características de los pasajeros que se salvaron del hundimiento del Titanic. Una de ellas es que con mayor probabilidad los pasajeros que viajaron en clase 1^a se salvaron, esto evidentemente influyó en la tarifa del billete que fueron las más cara de adquirir.

También hemos observado que los pasajeros con mayor edad viajaron en clases de mayor categoría.

Por otro lado, gracias a las gráficas barplot hemos podido observar que hay una mayor posibilidad de que si eras mujer te salvaras, esto nos lleva a concluir que las mujeres que viajaron en primera clase fueron las que tuvieron una mayor probabilidad de salvarse.

7. Código. (2 puntos)

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.

El código utilizado para obtener los resultados comentados se puede encontrar en la ruta “/davidarbona/practica2Titanic/src”

8. Referencias

- Calvo M, Subirats L, Pérez D (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
- Github Guides: Mastering Markdown. <https://guides.github.com/features/mastering-markdown/> (2014)
- Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
- Kaggle Inc. <https://www.kaggle.com/c/titanic>
- 2021 Stack Exchange Inc. <http://stackoverflow.com>
- 1996-2021 Encyclopedia Titanica. <https://www.encyclopedia-titanica.org/>