PRÁCTICA 2 - TIPOLOGÍA Y CICLO DE VIDA DE LOS DATOS

Autores: Barco Sousa, José Manuel y Velázquez Carricondo, Juan Antonio

Junio 2022

DESCRIPCIÓN DEL DATASET

Para la realización de esta práctica vamos a utilizar el dataset propuesto en el enunciado de la misma: Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic) En la web referida hay disponibles tres archivos:

- train: con un juego de datos con 12 variables
- test: con un juego de datos de 11 variables. Omite la correspondiente a la información de superviviencia del pasajero ya que la competición propuesta en la web se basa en predecir la superviviencia de los pasajeros recogidos en este archivo.
- gender_submission: para informar de las predicciones realizadas.

Así pues, el archivo que vamos a utilizar para la realización de esta práctica es train.csv, test.csv no nos es útil porque no nos permitirá chequear la precisión de las predicciones realizadas.

OBJETIVO

El objetivo que se plantea es determinar qué factores son los que tienen más incidencia en la probabilidad de supervivencia de los pasajeros

LECTURA DEL ARCHIVO

```
data<-read.csv("./train.csv",header=T,sep=",")
attach(data)
```

Vamos a realizar una primera aproximación al dataset.

```
str(data)
```

```
## '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" ...
```

Como podemos ver, contiene 891 registros de 12 variables.

Atendiendo a la información mostrada y la diccionario que se proporciona en la web, las variables recogen la siguiente información:

- PassengerId: corresponde con la identificación del pasajero/registro en el dataset
- Survived: recoge la información acerca de si el pasajero sobrevivió (1) o no (0) al naufragio
- PClass: Aunque es de tipo int es una variable categórica que recoge información acerca de la clase en que se embarcó el pasajero
- Name:recoge el nombre del pasajero
- Sex: recoge el sexo del pasajero
- Age:Recoge la edad del pasajero
- SibSp: Recoje información acerca del número hermanos y/o cónyuges del pasajero a bordo
- Parch: Recoje información acerca del número padres o hijos del pasajero a bordo
- Ticket: Recoje el número de ticket del pasajero
- Fare: Recoje el importe correspondiente a la tarifa abonada por el pasajero
- Cabin: Recoje el número de camarote del pasajero
- Embarked: Recoje el puerto de embarque del pasajero

LIMPIEZA Y TRANSFORMACIÓN DE DATOS

VALORES PERDIDOS

Vamos a comenzar por ver si existen valores perdidos en el dataset

```
colSums(is.na(data))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

```
colSums(data=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

Por la información obtenida, podemos ver que hay

- 177 registros perdidos en el campo Age.
- 687 en el campo Cabin
- 2 en el campo Embarked

Tratamiento de los valores perdidos

Dado el número de valores faltantes, creemos que lo mejor es prescindir de la variable Cabin. Quizás los números (pares o impares) o la letra que precede al número nos podría dar alguna información acerca de en qué lado del barco o en qué piso se encontraba la cabina, por si pudiese tener relación con la supervivencia, pero al disponer de tan pocos registros (204 de 891) con esta información creemos que es mejor prescindir de esta variable en comparación con la opción de no utilizar los registros con el valor perdido.

```
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
```

```
data<-select(data, -Cabin)
```

Por lo que respecta a la columna Age, por la información recogida en la web donde estaba el archivo del dataset, sabemos que ya hay valores estimados, son aquellos que contienen 0.5

```
table(data$Age)
```

```
##
## 0.42 0.67 0.75 0.83 0.92    1    2    3    4    5    6    7    8    9   10   11
##    1    1    2    2    1    7   10    6   10    4    3    3    4    8    2    4
##   12   13   14 14.5   15   16   17   18   19   20 20.5   21   22   23 23.5   24
##    1    2    6    1    5   17   13   26   25   15    1   24   27   15    1   30
## 24.5   25   26   27   28 28.5   29   30 30.5   31   32 32.5   33   34 34.5   35
##    1   23   18   18   25    2   20   25    2   17   18    2   15   15    1   18
##   36 36.5   37   38   39   40 40.5   41   42   43   44   45 45.5   46   47   48
##   22    1    6   11   14   13    2    6   13    5    9   12    2    3    9    9
##   49   50   51   52   53   54   55 55.5   56   57   58   59   60   61   62   63
##    6   10    7    6    1    8    2    1    4    2    5    2    4    3    4    2
##   64   65   66   70 70.5   71   74   80
##    2    3    1    2    1    2    1    1
```

Como podemos ver, la estimación no es siempre la misma, por lo que descartamos que sea algún valor central de la muestra.

Para hacer la imputación de valores perdidos en esta variable, utilizaremos la función mice, a la que indicamos las variables a tomar en consideración para realizar las estimaciones y que nos permite elegir el método para estimar el valor a imputar: "mean" para utilizar la media, "norm.boot" para regresión lineal usando bootstrap, "cart" para utilizar árboles de decisión. "rf" para utilizar randomforest, etc...

Tras la aplicación de la función, sustituimos los valores NA de Age por los estimados con mice.

```
library(mice)
```

```
##
## Attaching package: 'mice'
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
para_imiputar <- mice(data%>%select(Survived, Pclass, SibSp, Parch, Age), method = "cart")
```

```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
##   2   4  Age
##   2   5  Age
##   3   1  Age
##   3   2  Age
##   3   3  Age
##   3   4  Age
##   3   5  Age
##   4   1  Age
##   4   2  Age
##   4   3  Age
##   4   4  Age
##   4   5  Age
##   5   1  Age
##   5   2  Age
##   5   3  Age
##   5   4  Age
##   5   5  Age
```

```
imputacion <- mice::complete(para_imiputar)
data<-data%>%mutate(Age =imputacion$Age)
colSums(is.na(data))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch      Ticket        Fare    Embarked
##           0           0           0           0           0
```

Como podemos comprobar, ya no hay valores perdidos en Age

Variables a desechar

Creemos también que hay otra serie de variables que no tienen relación con la supervivencia. Estas son:

- PassengerId
- Name
- Embarked
- Ticket

No parece que ni la identidad del pasajero, ni el puerto donde haya embarcado, ni su número de ticket influyan en su probabilidad de supervivencia.

```
data<-select(data, -PassengerId, -Name, -Embarked, -Ticket)
```

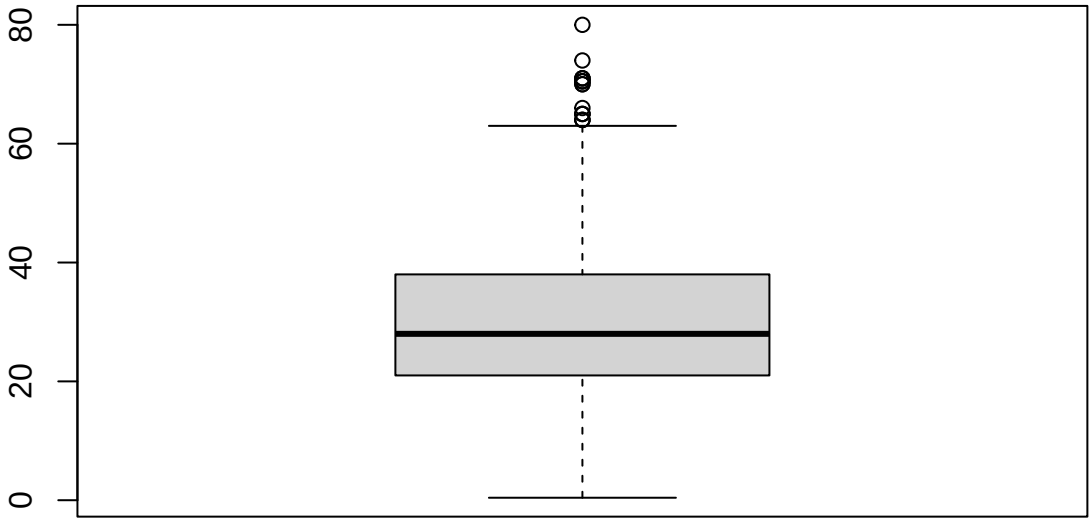
VALORES EXTREMOS

Dados los campos disponibles en el dataset, sólo tiene sentido buscar valores extremos en las siguientes variables, que son la numéricas:

- Age
- SibSP
- Parch
- Fare

Vamos a ver los boxplots correspondientes y a identificar los valores extremos de estos campos

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

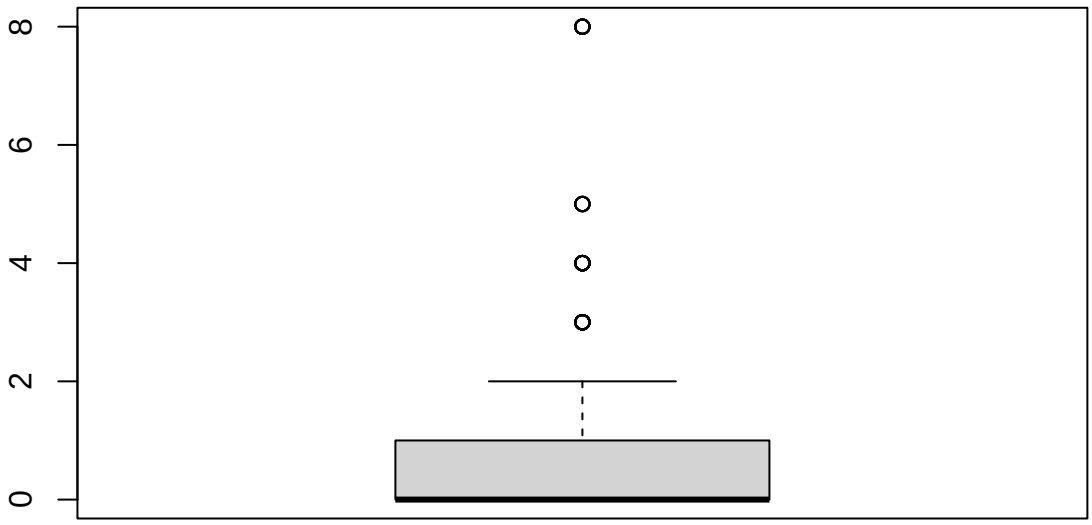



```
boxplot.stats(data$Age)$out
```

```
##  [1] 66.0 65.0 71.0 70.5 70.5 65.0 64.0 65.0 71.0 71.0 64.0 64.0 80.0 70.0 70.0
## [16] 70.5 74.0
```

Como puede apreciarse, estos valores extremos, pueden ser valores lícitos, ya que son edades posibles.

```
boxplot(data$SibSp)
```

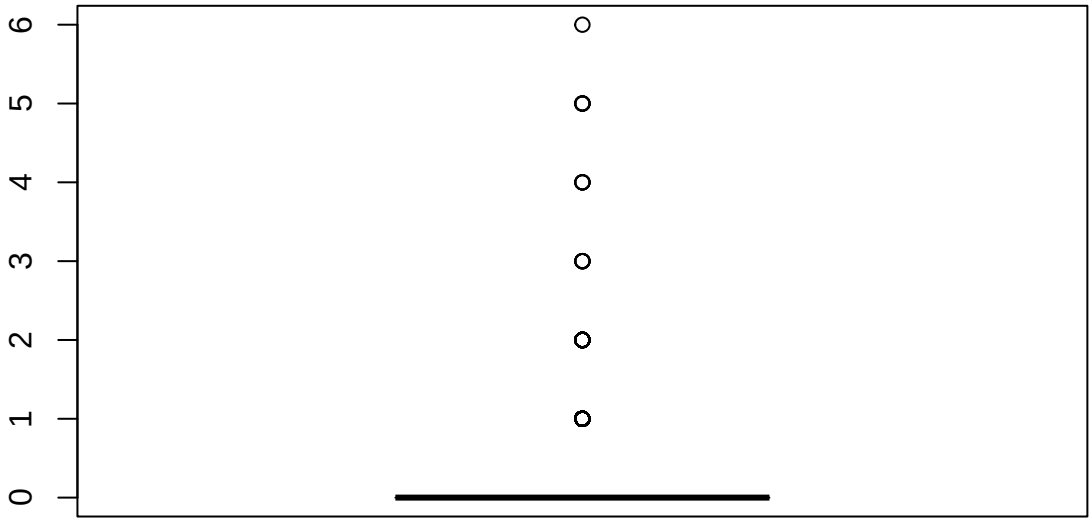



```
boxplot.stats(data$SibSp)$out
```

```
##  [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
```

Como puede apreciarse, estos valores extremos, pueden ser valores lícitos. La mayoría del pasaje tenía a bordo ningún o 1 hermano o cónyuge, pero puede ser que familias numerosas se embarcasen en un viaje familiar dando como resultado que tuviesen en el barco 8 familiares entre hermanos o cónyujes

```
boxplot(data$Parch)
```

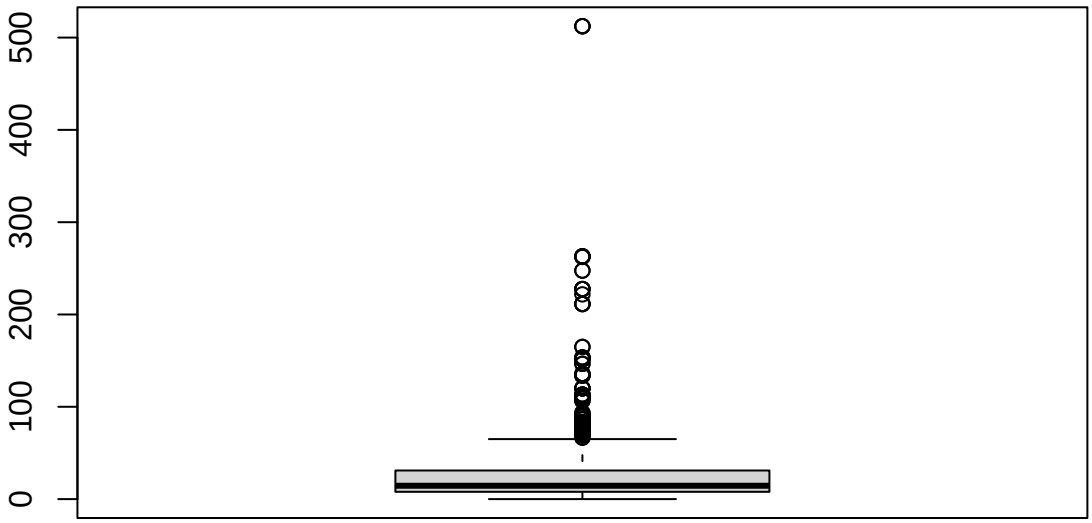



```
boxplot.stats(data$Parch)$out
```

```
##   [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1 2 1
##  [38] 1 2 1 4 1 1 1 1 2 2 1 2 1 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 1 1 1 1 1 1 2 1 2
##  [75] 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2 2 3 4 1 2 1
## [112] 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1 2 1 1 2 5 2 1 1
## [149] 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1 1 2 1 2 3 1 2 1 2 2
## [186] 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1 1 5 2
```

Como puede apreciarse, estos valores extremos, pueden ser valores lícitos. La reflexión sobre la anterior variable hace que pueda pensarse en pasajeros con 5 0 6 familiares a bordo, en este caso o hijos y padres.

```
boxplot(data$Fare)
```




```
boxplot.stats(data$Fare)$out
```

```
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000
##   [9] 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000  66.6000  69.5500
##  [17]  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000  83.4750  90.0000
##  [25]  79.2000  86.5000 512.3292  79.6500 153.4625 135.6333  77.9583  78.8500
##  [33]  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
##  [49] 263.0000  75.2500  69.3000 135.6333  82.1708 211.5000 227.5250  73.5000
##  [57] 120.0000 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042  91.0792
##  [65]  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000 221.7792 106.4250
##  [73]  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
##  [89]  73.5000 512.3292  76.7292 211.3375 110.8833 227.5250 151.5500 227.5250
##  [97] 211.3375 512.3292  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583
## [105] 211.3375  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500
## [113]  89.1042 164.8667  69.5500  83.1583
```

Como puede apreciarse, estos valores extremos, pueden ser valores lícitos. La mayoría de los pasajeros pagaron tarifas más baratas, pero no es irracional pensar que una minoría de pasajeros pudieran pagar tarifas sensiblemente más altas por un servicio diferencial.

Por otra parte, también podría ser que en Fare se recogiera el importe total pagado por el pasaje de un grupo de personas, en aquellos casos en que viajan a bordo varios miembros de una misma familia.

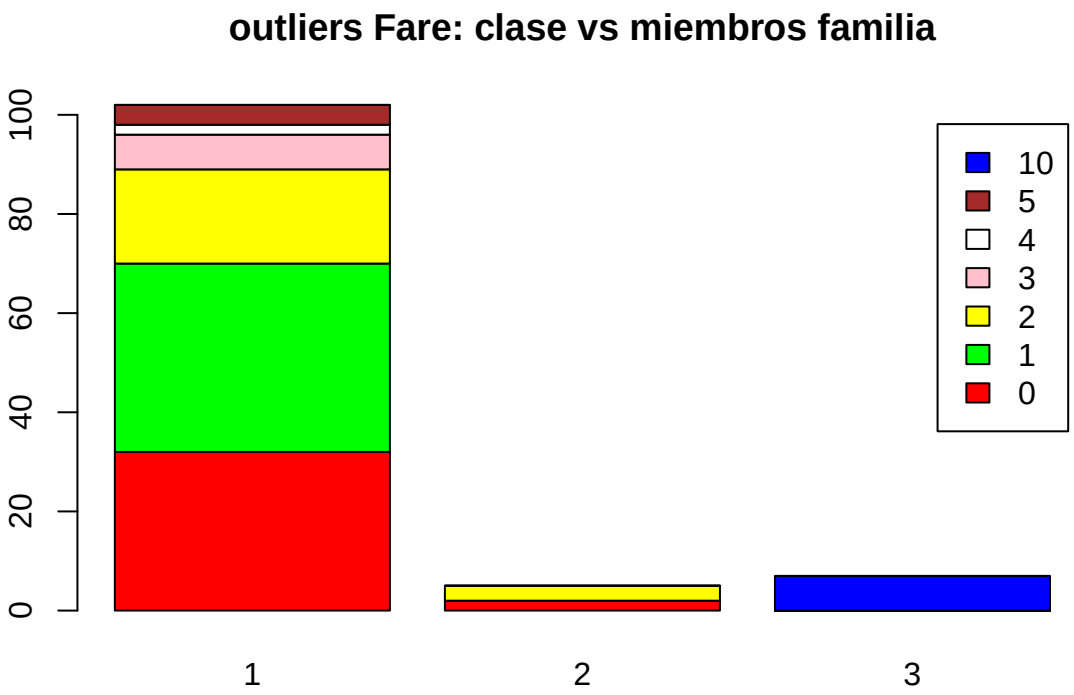
Vamos a intentar ver a través de una visualización cual es el caso.

Para ello, primero, creamos una nueva variable con la suma de los familiares a bordo de cada pasajero.

A continuación, utilizando sólo los registros correspondientes a outliers de Fare, relacionamos clase y número de miembros de la familia.

```
data$fam=data$SibSp+data$Parch
barplot(table(
  data$fam[data$Fare>min(boxplot.stats(data$Fare)$out)],
  data$Pclass[data$Fare>min(boxplot.stats(data$Fare)$out)]),
  col = c("red", "green", "yellow", "pink", "white", "brown", "blue", "orange", "purple", "grey"),
  main="outliers Fare: clase vs miembros familia",
```

```
  legend.text = rownames(
    table(data$fam[data$Fare>min(boxplot.stats(data$Fare)$out)],
    data$Pclass[data$Fare>min(boxplot.stats(data$Fare)$out)]))
  )
```



Como podemos apreciar en los gráficos, parece que puede ser la primera opción expuesta, ya que esos valores más altos de Fare corresponden mayoritariamente con billetes de primera clase.

Además, en el gráfico vemos que la mayoría de esos valores, corresponden a familias de entre 0 y 2 miembros que viajaban en primera clase.

Así pues, no parece probable que el importe responda al abono total por los pasajes de todos los miembros de una misma familia. Sino más bien a que, como dijimos, correspondan a tarifas más altas por algún servicio diferencial de algún tipo.

TRANSFORMACIÓN DE DATOS

Vamos a crear una nueva variable con la discretización de la variable Age

```
data$Edad<-cut(data$Age,
               breaks = c(0,16,36,60,max(data$Age)+1),
               labels = c("infancia", "juventud", "madurez", "vejez"))
```

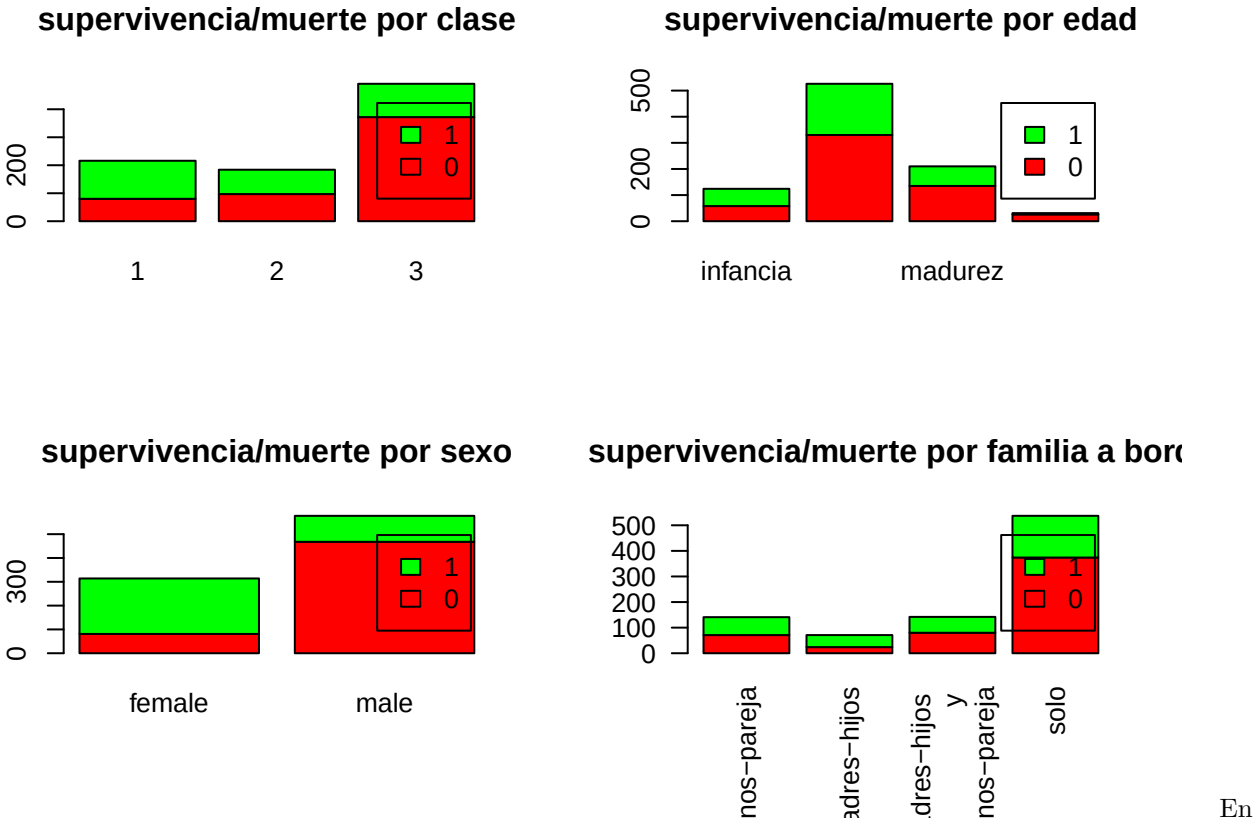
Ahora vamos a crear una nueva variable para saber con quien viaja cada pasajero.

```
data<-mutate(data, Familia=case_when(
      SibSp==0 & Parch==0 ~ "solo",
      SibSp==0 & Parch>0 ~ "padres-hijos",
      SibSp>0 & Parch==0 ~ "hermanos-pareja",
      SibSp>0 & Parch>0 ~ "padres-hijos \n y \n hermanos-pareja"))
```

ANÁLISIS DE LOS DATOS

Antes de comenzar con los análisis, vamos a utilizar las últimas transformaciones realizadas para hacer una inspección visual de las posibles relaciones de las variables que tenemos con la supervivencia

```
par(mfrow = c(2,2))
barplot(table(data$Survived, data$Pclass),
        col=c("red", "green"),
        main="supervivencia/muerte por clase",
        legend.text = rownames(table(data$Survived, data$Pclass)))
barplot(table(data$Survived, data$Edad),
        col=c("red", "green"),
         main="supervivencia/muerte por edad",
        legend.text = rownames(table(data$Survived, data$Edad)))
barplot(table(data$Survived, data$Sex),
        col=c("red", "green"),
         main="supervivencia/muerte por sexo",
        legend.text = rownames(table(data$Survived, data$Sex)))
barplot(table(data$Survived, data$Familia),
        col=c("red", "green"),
         main="supervivencia/muerte por familia a bordo",
        legend.text =rownames(table(data$Survived, data$Familia)),
        las=2)
```



En esta primera aproximación visual, ya podemos ver que:

- la supervivencia se reduce en función de la clase, proporcionalmente, sobrevivieron muchos más pasajeros de primera de que tercera

- proporcionalmente, la supervivencia también fue mayor entre los pasajeros más jóvenes
- proporcionalmente, también sobrevivieron más las mujeres que los hombres
- proporcionalmente, la supervivencia fue sensiblemente menor entre los que viajaban sólos y mayor entre los que viajaban con sus padres y/o hijos

COMPROBACIÓN DE NORMALIDAD

Vamos a comprobar la normalidad de las variables numéricas que tenemos

```
shapiro.test(data$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Age
## W = 0.97812, p-value = 2.644e-10
```

```
shapiro.test(data$SibSp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$SibSp
## W = 0.51297, p-value < 2.2e-16
```

```
shapiro.test(data$Parch)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Parch
## W = 0.53281, p-value < 2.2e-16
```

```
shapiro.test(data$Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Fare
## W = 0.52189, p-value < 2.2e-16
```

Cómo podemos observar por los p-values (menores a 0,05), ninguna de ellas cumple la condición de normalidad

COMPROBACIÓN DE HOMOGENEIDAD DE LA VARIANZA

Vamos ahora a comprobar la homogeneidad de la varianza de esas mismas varibles

```
fligner.test(Age ~ SibSp, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by SibSp
## Fligner-Killeen:med chi-squared = 24.074, df = 6, p-value = 0.0005062
```

```
fligner.test(Age ~ Parch, data = data)
```

```
##
```

```
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Parch
## Fligner-Killeen:med chi-squared = 57.118, df = 6, p-value = 1.729e-10
```

```
fligner.test(Age ~ Fare, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Fare
## Fligner-Killeen:med chi-squared = 307.32, df = 247, p-value = 0.005396
```

```
fligner.test(SibSp ~ Parch, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  SibSp by Parch
## Fligner-Killeen:med chi-squared = 198.79, df = 6, p-value < 2.2e-16
```

```
fligner.test(SibSp ~ Fare, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  SibSp by Fare
## Fligner-Killeen:med chi-squared = 329.55, df = 247, p-value = 0.0003427
```

```
fligner.test(Parch ~ Fare, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Parch by Fare
## Fligner-Killeen:med chi-squared = 313.56, df = 247, p-value = 0.00263
```

En todos los casos el p-value es inferior a 0,05 por lo que concluimos que las varianzas no son homogéneas

PRUEBAS ESTADÍSTICAS

chi square test

En primer lugar vamos a confirmar si las afirmaciones que hicimos a la vista de los gráficos al comienzo de este apartado son ciertas, es decir, si hay relación entre la clase, el sexo, la edad y la Familia a bordo con la supervivencia

```
chisq.test(data$Survived, data$Pclass)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data$Survived and data$Pclass
## X-squared = 102.89, df = 2, p-value < 2.2e-16
```

```
chisq.test(data$Survived, data$Sex)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
```

```
##
## data:  data$Survived and data$Sex
## X-squared = 260.72, df = 1, p-value < 2.2e-16
```

```
chisq.test(data$Survived, data$Edad)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data$Survived and data$Edad
## X-squared = 18.954, df = 3, p-value = 0.0002795
```

```
chisq.test(data$Survived, data$Familia)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data$Survived and data$Familia
## X-squared = 47.098, df = 3, p-value = 3.314e-10
```

Como puede deducirse de los p-values, en todos los casos el test es significativo, es decir, las variables están correlacionadas.

Modelo lineal

Ahora vamos a intentar hacer predicciones.

Inicialmente utilizamos un modelo lineal.

En primer lugar, dividiremos nuestros datos para entrenamiento y test

```
library(caret)
```

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

```
## Loading required package: lattice
```

```
set.seed(987654321)
trainIndex=createDataPartition(data$Survived, p=0.80)$Resample1

data_train=data[trainIndex, ]
data_test=data[-trainIndex, ]
```

Ahora Vamos a entrenar un modelo para, después, hacer predicciones sobre los datos de test

```
clasificadorRL <- glm(Survived~Age+SibSp+Parch+Fare+Pclass+Sex, family = binomial, data = data_train)
print("************MODELO************")
```

```
## [1] "************MODELO************"
```

```
summary(clasificadorRL)
```

```
##
## Call:
## glm(formula = Survived ~ Age + SibSp + Parch + Fare + Pclass +
##     Sex, family = binomial, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7992  -0.5853  -0.3769   0.5989   2.5649
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.391392   0.627149   8.597  < 2e-16 ***
## Age         -0.045097   0.008686  -5.192 2.08e-07 ***
## SibSp       -0.531288   0.132905  -3.997 6.40e-05 ***
## Parch       -0.042116   0.140310  -0.300    0.764
## Fare         0.002517   0.002471   1.018    0.309
## Pclass      -1.169438   0.161586  -7.237 4.58e-13 ***
## Sexmale     -2.868701   0.229356 -12.508  < 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: 938.80  on 712  degrees of freedom
## Residual deviance: 608.41  on 706  degrees of freedom
## AIC: 622.41
##
## Number of Fisher Scoring iterations: 5
```

```
pred_test <- predict(clasificadorRL, type = 'response', newdata = data_test)
pred_test <- ifelse(pred_test>0.5, 1, 0)
pred_test <- factor(pred_test, levels = c("0", "1"))
matrizConfusion <- table(data_test$Survived, pred_test)
print("************MATRIZ DE CONFUSION************")
```

```
## [1] "************MATRIZ DE CONFUSION************"
```

```
matrizConfusion
```

```
##    pred_test
##      0  1
##   0 87 12
##   1 24 55
```

```
print(paste("porcetaje de casos bien clasificados: ", 100*(matrizConfusion[1,1]+matrizConfusion[2,2])/su
```

```
## [1] "porcetaje de casos bien clasificados:  79.7752808988764"
```

Como se pude ver, el modelos estimado el valor de la constante (intercept) sumado a cada una de las variable multiplicada por el parámetro estimado por el modelo (Estimate). Como se puede ver, salvo Fare, todos los parámetros son negativos. Hay que señalar que las variables Parch y Fare no son significativas.

Como también podemos ver, el modelo clasifica bien el 78% de los casos, que no es un porcentaje demasiado bueno.

Árbol de decisión

Vamos, ahora, a probar con un árbol de decisión.

Para ello partiremos nuestros datos de entrenamiento y test para tener por un lado la variable objetivo (la supervivencia y, por el otro, el resto de variables).

A continuación, entrenamos el modelo y usamos los datos de test para predecir y calcular el porcentaje de clasificaciones correctas.

```
library(C50)
data_train_x<-select(data_train, -Survived, -Familia, -Edad)
data_train_y<-as.factor(data_train$Survived)
```

```
data_test_x<-select(data_test, -Survived, -Familia, -Edad)
data_test_y<-as.factor(data_test$Survived)
model <- C50::C5.0(data_train_x, data_train_y,rules=TRUE )
predicted_model <- predict( model, data_test_x, type="class" )
matrizConfusion2 <- table(data_test_y, predicted_model)
matrizConfusion2
```

```
##            predicted_model
## data_test_y  0  1
##           0 87 12
##           1 19 60
```

```
print(paste("porcetaje de casos bien clasificados: ", 100*(matrizConfusion2[1,1]+matrizConfusion2[2,2]),
```

```
## [1] "porcetaje de casos bien clasificados:  82.5842696629213"
```

Como se puede ver, el modelo generado con el árbol de decisión mejora el anterior. Este clasifica clasifica correctamente el 80% de los casos.

En el siguiente listado vamos a ver el porcentaje de observaciones de entrenamiento que caen en todos los nodos generados tras una división en el que ha participado la variable

```
C5imp(model, metric = "usage")
```

```
##        Overall
## Sex      94.95
## Pclass   57.64
## Age      53.72
## Fare     38.43
## SibSp    30.72
## fam      26.65
## Parch    10.94
```

Cómo podemos ver bajo la variable Sex, caen el 84,43% de los casos. Bajo decisiones que implican a la variable Age caen el 79,94% de los caos. Sobre PClass caen el 67,46 de los casos y bajo SibSp el 57,22%.

Como pasaba con el modelo anteior, Parch y Fare no son significativas, ningún caso cae bajo una decisión en la que estuviesen implicadas.

Ahora vamos a ver el porcentaje de decisiones en que participa cada variable

```
C5imp(model, metric = "splits")
```

```
##          Overall
## Age    25.000000
## Pclass 19.444444
## Fare   16.666667
## Sex    16.666667
## SibSp  11.111111
## fam     8.333333
## Parch   2.777778
```

A mayores, podemos también ver las reglas generadas por el modelo

```
summary(model)
```

```
##
## Call:
## C5.0.default(x = data_train_x, y = data_train_y, rules = TRUE)
##
```

```
##
## C5.0 [Release 2.07 GPL Edition]      Thu May 19 14:05:49 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 713 cases (8 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (43/3, lift 1.4)
##  Pclass > 2
##  fam > 3
##  ->  class 0  [0.911]
##
## Rule 2: (21/2, lift 1.4)
##  Pclass > 2
##  Age > 16
##  SibSp > 0
##  Fare <= 15
##  ->  class 0  [0.870]
##
## Rule 3: (98/13, lift 1.4)
##  Pclass > 2
##  Age > 30
##  SibSp <= 0
##  ->  class 0  [0.860]
##
## Rule 4: (78/12, lift 1.3)
##  Pclass > 2
##  Age > 16
##  Age <= 30
##  Parch <= 0
##  Fare > 7.925
##  ->  class 0  [0.838]
##
## Rule 5: (471/84, lift 1.3)
##  Sex = male
##  ->  class 0  [0.820]
##
## Rule 6: (125/5, lift 2.6)
##  Pclass <= 2
##  Sex = female
##  ->  class 1  [0.953]
##
## Rule 7: (15, lift 2.6)
##  Sex = male
##  Age <= 13
##  SibSp <= 2
##  ->  class 1  [0.941]
##
## Rule 8: (130/13, lift 2.4)
##  Sex = female
##  Fare > 15
```

```
##  fam <= 3
##  ->  class 1  [0.894]
##
## Rule 9: (39/5, lift 2.3)
##  Sex = female
##  Age <= 16
##  fam <= 3
##  ->  class 1  [0.854]
##
## Rule 10: (85/18, lift 2.1)
##  Sex = female
##  Age <= 30
##  SibSp <= 0
##  ->  class 1  [0.782]
##
## Rule 11: (25/5, lift 2.1)
##  Pclass <= 1
##  Age <= 45
##  Fare > 26
##  Fare <= 37.0042
##  ->  class 1  [0.778]
##
## Rule 12: (108/27, lift 2.0)
##  Pclass <= 1
##  Age <= 45
##  Fare > 26
##  ->  class 1  [0.745]
##
## Default class: 0
##
##
## Evaluation on training data (713 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##      12   88(12.3%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     421    29    (a): class 0
##      59   204    (b): class 1
##
##
##  Attribute usage:
##
##   94.95% Sex
##   57.64% Pclass
##   53.72% Age
##   38.43% Fare
##   30.72% SibSp
##   26.65% fam
```

```
##   10.94% Parch
##
##
## Time: 0.0 secs
```

En cuanto a estas reglas vemos que:

- según la primera, viajando en tercera clase y siendo mayor de 38, con una validez del 89,5%, la clasificación será que muere (clase 0)
- según la segunda, siendo hombre mayor de 13 años, con una validez del 84,5%. la clasificación será muere

...

- según la regla 4, viajando en primera o segunda clase y siendo mujer, con una validez del 95,3%. la clasificación será spbrevive

...

CONCLUSIONES

A lo largo del desarrollo de la práctica

- en primer lugar, hemos intentado, ir conociendo los datos: las variables que incluía el dataset y sus tipos
- hemos aproximado también a los valores que incluía cada variables y hemos detectado valores problemáticos (perdidos y outliers) para decidir que hacemos con ellos. De hecho hemos desechado alguna variable por contener perdidos y hemos imputado valores en otra de ellas. Respecto a los outliers, los hemos identificado e intentado comprender, si eran lícitos o no, intentado en un caso en concreto, el de Fare, buscar una explicación a sus valores.
- hemos creado nuevas variables a partir de las existentes para facilitar la tarea de encontrar relaciones entre ellas a través de la visualización de los datos.
- además de la visualización, hemos realizado test estadísticos para verificar si confirmaban las conclusiones que sacamos del análisis de los gráficos.
- finalmente, hemos intentado hacer predicciones utilizando dos herramientas diferentes: glm, ya que los datos no pasaron las pruebas de normalidad y homocedasticidad y mediante árboles de decisión.

En el caso de la predicción hemos visto que, para estos datos, los dos métodos alcanzan porcentajes de acierto muy similares y que de ambos se sacan conclusiones muy parecidas:

- hay variables que no tienen relación con la superviviencia. El modelo glm les asigna 0 como valor del estimador a Parch y Fare, además de mostrar que no son estadísticamente significativas. En el caso del árbol de decisión no son utilizadas en ningún corte.
- Los dos modelos identifican como variables relacionadas con la supervivencia la clase, el sexo (el sexo masculino relacionado negativamente con la supervivencia), la edad y SibSp, aunque a la vista de los estimados y los cortes en que son utlizados y los casos que caen bajo cortes en los que intervienen, difieren un poco en el orden de importancia.