Tipología y ciclo de vida de los datos: PRA2

Autores: Marta Rodríguez y Pedro Félez

Diciembre 2020

1 Descripción del dataset

El objetivo de esta práctica, será el realizar un análisis del dataset que contiene información sobre los pasajeros que viajaban en el titanic. Estos conjuntos de datos se han tomado de la página web de Kaggle.

Este conjunto de datos es muy utilizado en la ciencia de datos, ya que es sencillo de entender (todos hemos oido hablar sobre el Titanic) y permite realizar algoritmos predictivos sobre la variable Survived. Es decir, permite generar modelos predictivos que nos indiquen si un pasajero sobrevivio o no al Titanic en función de otras características del conjunto de datos.

Este dataset se encuentra dividido en 2 partes, una para el entrenamiento del algoritmo y otra para las prueba. Nosotros los vamos a unir para tener el conjunto completo y así poder tratar y seleccionar los datos que nos interesan ya que no se va a utilizar todas las variables para el análisis.

El primer paso para realizar el análisis es cargar los conjuntos de datos desde los archivos csv en los que se encuentran.

```
library(ggplot2)
library(dplyr)
datosTitanicTrain <- read.csv('train.csv',stringsAsFactors = FALSE, header =
TRUE, strip.white = TRUE)
datosTest <- read.csv('test.csv',stringsAsFactors = FALSE, header = TRUE, st
rip.white = TRUE)
datosSubmision <- read.csv('gender_submission.csv',stringsAsFactors = FALSE,
header = TRUE, strip.white = TRUE)
```

Observamos que los datos estan divididos en tres subconjuntos, el conjunto de entrenamiento (datosTitanicTrain). Por otro lado, se encuentra el conjunto de validación (daosTest) donde la variable que indica si un pasajero sobrevive o no se encuentra en datosSubmision.

2 Integración y selección de los datos a analizar

El primer paso en el proceso de análisis es realizar un análisis descriptivo sobre el total del conjunto de datos. Por lo tanto, se ha decidido agrupar la información en un unico dataset, que nos permita tener una

visión lo más general posible de los datos. En el momento dado, cuando se tengan que realizar modelos predictivos, se volverá a dividir este dataset para poder validar el modelo con un conjunto de datos de validación.

En el primer paso de esta integración de datos vamos a unir los dos dos conjuntos de datos de validación, para ello añadimos la columna Survived al dataset datosTest con los valores de datosSubmision. Se toma el nombre de Survived en esta variable para que los nombres de las columnas de datosTitanicTrain y datosTest coincidan.

```
#Añadimos la columna survived
datosTest["Survived"] <- datosSubmision["Survived"]
head(datosTest)
```

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

```
head(datosTitanicTrain)
```

```
##   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 Pa
rch
## 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 observa que el orden de las columnas de los datasets datosTitanicTrain y datosTest no coinciden, por lo que se ponen en el mismo orden para posteriormente poder unir ambos datasets.

```
#Ordenamos las columnas
datosTest = datosTest[, c(1,12,2,3,4,5,6,7,8,9,10,11)]
#Unimos los datasets
datosTitanic <- rbind(datosTitanicTrain, datosTest)
```

Comprobamos que la integración de los datasets se ha realizado correctamente, para ello mostramos los primeros registros y los últimos.

```
head(datosTitanic)
```

```
##   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 Pa
rch
## 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
```

```
tail(datosTitanic)
```

```
##      PassengerId Survived Pclass                          Name    Sex  A
ge
## 1304        1304        1      3 Henriksson, Miss. Jenny Lovisa female 2
8.0
## 1305        1305        0      3             Spector, Mr. Woolf   male
NA
## 1306        1306        1      1   Oliva y Ocana, Dona. Fermina female 3
9.0
## 1307        1307        0      3   Saether, Mr. Simon Sivertsen   male 3
8.5
## 1308        1308        0      3            Ware, Mr. Frederick   male
NA
## 1309        1309        0      3       Peter, Master. Michael J   male
NA
##      SibSp Parch             Ticket     Fare Cabin Embarked
## 1304     0     0             347086   7.7750             S
## 1305     0     0          A.5. 3236   8.0500             S
## 1306     0     0           PC 17758 108.9000  C105        C
## 1307     0     0 SOTON/O.Q. 3101262   7.2500             S
## 1308     0     0             359309   8.0500             S
## 1309     1     1               2668  22.3583             C
```

Vemos que el conjunto de datos esta formado por 13 variables que nos indica distinta información sobre el pasajero, veamos que información contiene cada una de estas variables:

- PassengerId: Indica el identificador de cada uno de los pasajeros.
- Survived: Variable que indica si sobrevive 1, o si por el contrario no sobrevive al accidente 0.
- Pclass: Variable con la clase del tiquet 1 = 1st, 2 = 2nd, 3 = 3rd
- Name: Nombre del pasajero.
- Sex: Sexo del pasajero (female/male).
- Age: Edad del pasajero en años.
- SibSp: Número de hermanos/esposas en el Titanic.
- Parch: Número de padres/hijos en el Titanic.
- Ticket: Número de ticket del pasajero.

- Fare: Tarifa del pasajero.
- Cabin: Número de cabina del pasajero.
- Embarked: Indica el puerto de embarque C = Cherbourg, Q = Queenstown, S = Southampton

3 Limpieza de los datos

Observamos que las variables Cabin, Fare, Ticket, Name y PassengerId son variables demasiado expecíficas de los pasajeros, algunas como el nombre, número de tiquet incluso específicas de cada uno. Por esta razón, se decide realizar una reducción de los datos, eliminando estas variables del conjunto de datos para realizar el análisis. Si se incluyeran entraríamos en una sobrexpecialización de los modelos que los complicarían y serían demasiado expecíficos para aplicarlos a nuevos datos.

```
datosTitanic <- datosTitanic[, -11]
datosTitanic <- datosTitanic[, -10]
datosTitanic <- datosTitanic[, -9]
datosTitanic <- datosTitanic[, -4]
datosTitanic <- datosTitanic[, -1]
```

Mediante las variables SibSp y Parch se puede saber si un pasajero viaja solo o acompañado. Por ello, se ha decidido realizar una nueva variable alone que nos indique directamente si el pasajero viaja solo o acompañado.

```
datosTitanic$alone <- with(datosTitanic, ifelse(SibSp==0 & Parch==0, "yes",
"no"))
```

Una vez tenemos las variables que nos interesan estudiar, se puede comenzar realizando un análisis descriptivo sobre el conjunto de datos.

```
str(datosTitanic)
```

```
## 'data.frame':    1309 obs. of  8 variables:
##  $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ 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 ...
##  $ Embarked: chr  "S" "C" "S" "S" ...
##  $ alone   : chr  "no" "no" "yes" "no" ...
```

```
head(datosTitanic)
```

```
##   Survived Pclass    Sex Age SibSp Parch Embarked alone
## 1        0      3   male  22     1     0        S    no
## 2        1      1 female  38     1     0        C    no
## 3        1      3 female  26     0     0        S   yes
```

```
## 4        1      1 female  35     1     0        S    no
## 5        0      3   male  35     0     0        S   yes
## 6        0      3   male  NA     0     0        Q   yes
```

```
summary(datosTitanic)
```

```
##     Survived          Pclass          Sex                 Age
##  Min.   :0.0000   Min.   :1.000   Length:1309        Min.   : 0.17
##  1st Qu.:0.0000   1st Qu.:2.000   Class :character   1st Qu.:21.00
##  Median :0.0000   Median :3.000   Mode  :character   Median :28.00
##  Mean   :0.3774   Mean   :2.295                      Mean   :29.88
##  3rd Qu.:1.0000   3rd Qu.:3.000                      3rd Qu.:39.00
##  Max.   :1.0000   Max.   :3.000                      Max.   :80.00
##                                                      NA's   :263
##      SibSp            Parch         Embarked            alone
##  Min.   :0.0000   Min.   :0.000   Length:1309        Length:1309
##  1st Qu.:0.0000   1st Qu.:0.000   Class :character   Class :character
##  Median :0.0000   Median :0.000   Mode  :character   Mode  :character
##  Mean   :0.4989   Mean   :0.385
##  3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :8.0000   Max.   :9.000
##
```

Tras realizar un primer análisis, se observan distintas cosas:

-Existe algún valor vacío en edad.

-El 37,7% de los pasajeros sobrevivieron.

-Las variables SibSp y Parch tienen muchos valores 0.

Se irá profundizando en el análisis para extraer mejores conclusiones.

Se observa que la variable edad se desconoce para 263 pasajeros. Mientras que la variable que indica el puerto de embarque del pasajero se desconoce en dos casos.

```
colSums(is.na(datosTitanic))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch Embarked    alone
##        0        0        0      263        0        0        0        0
```

```
colSums(datosTitanic=="")
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch Embarked    alone
##        0        0        0       NA        0        0        2        0
```

Como la variable Embarked es categórica, se completa con la etiqueta Desconocido para los valores perdidos.

```
datosTitanic$Embarked[datosTitanic$Embarked==""]="Desconocido"
```

Por otro lado, para la variable edad vamos a asignar valores a los registros perdidos. Para ser lo más precisos posible en esta asignación, se va a hacer uso de la función knnImputation, que a los valores perdidos le asignará un valor de edad según sus 10 registros más cercanos.

La función knnImputation solo acepta variables numéricas, ya que ha de calcular la distancia entre los registros. Por lo tanto, para realizar la estimación no se tienen en cuenta las variables categóricas sex, Embarked y Alone.

```
#Cargamos la libreria que contiene la función knnImputation
library(DMwR)
#Ejecutamos la función
new_data <- knnImputation(datosTitanic[, -c(3,7,8)], k = 10)
#Mostramos los resultados
colSums(is.na(new_data))
```

```
## Survived   Pclass      Age    SibSp    Parch
##        0        0        0        0        0
```

Como se puede observar, la variable Age ya no tiene valores perdidos, por lo tanto asignamos los valores de esta variable al dataset original

```
datosTitanic$Age <- new_data$Age
```

3.1 Identificación y tratamiento de valores extremos

Una vez se han analizado los valores perdidos, veamos si se observan valores extremos y si hubiera que tratarlos de algún modo. Veamos que las variables categóricas no toman ningún valor fuera de lo común.

```
unique(datosTitanic$Embarked)
```

```
## [1] "S"           "C"           "Q"           "Desconocido"
```

```
unique(datosTitanic$Survived)
```

```
## [1] 0 1
```

```
unique(datosTitanic$Pclass)
```

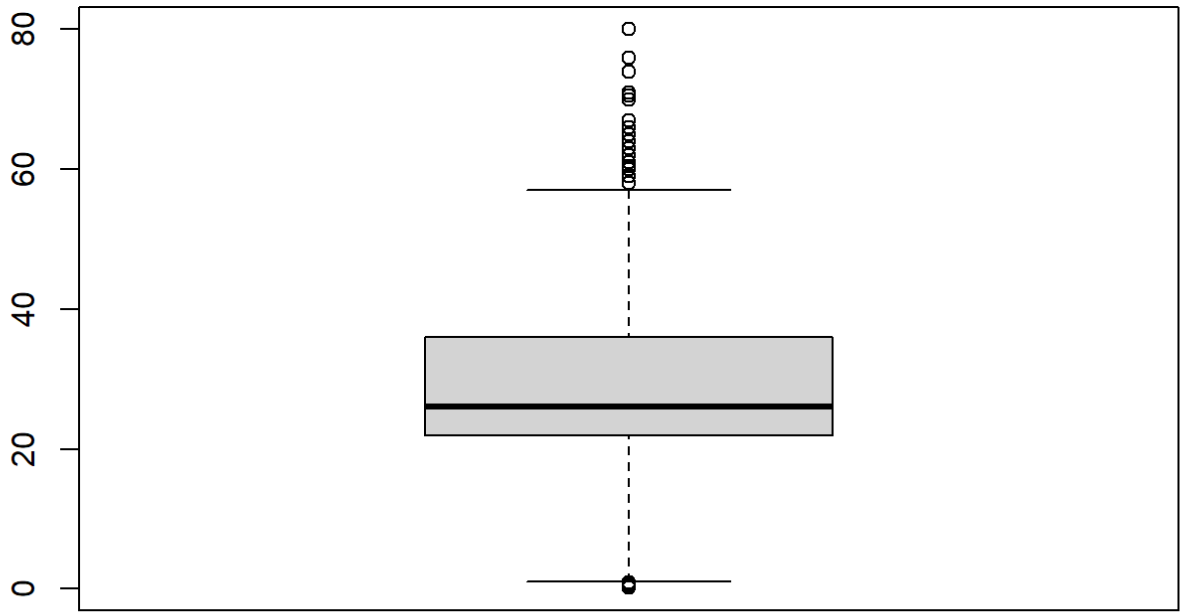
```
## [1] 3 1 2
```

```
unique(datosTitanic$Sex)
```

```
## [1] "male"   "female"
```

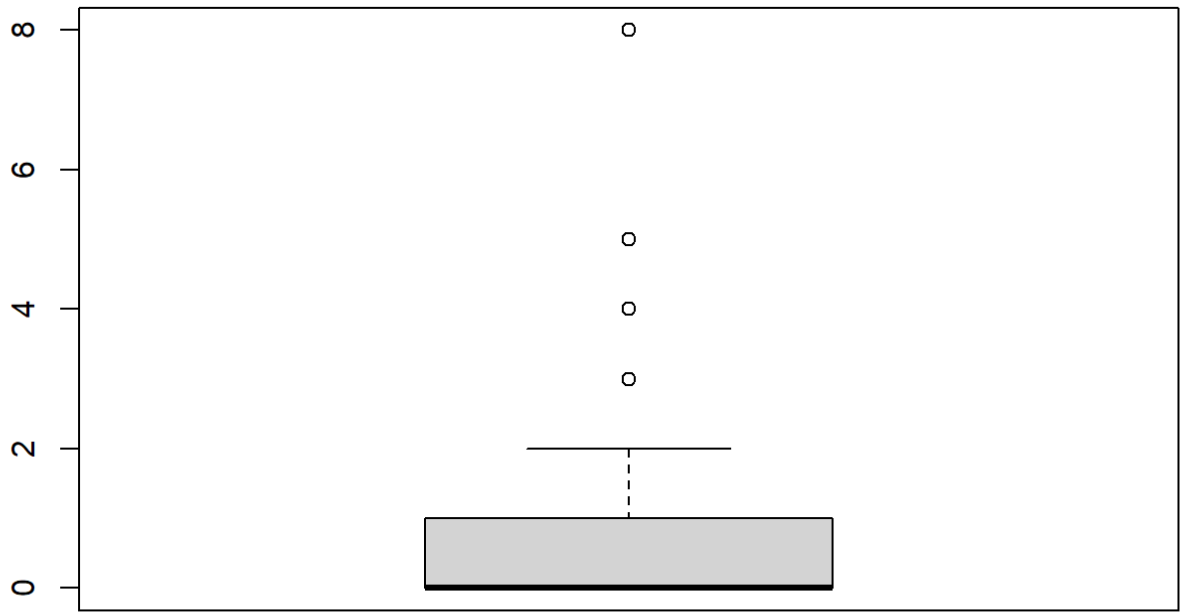
En ninguno de los casos se observan valores fuera de lo común. Analicemos los cuantiles de las variables numéricas para detectar posibles valores extremos.

```
boxplot(datosTitanic$Age)
```



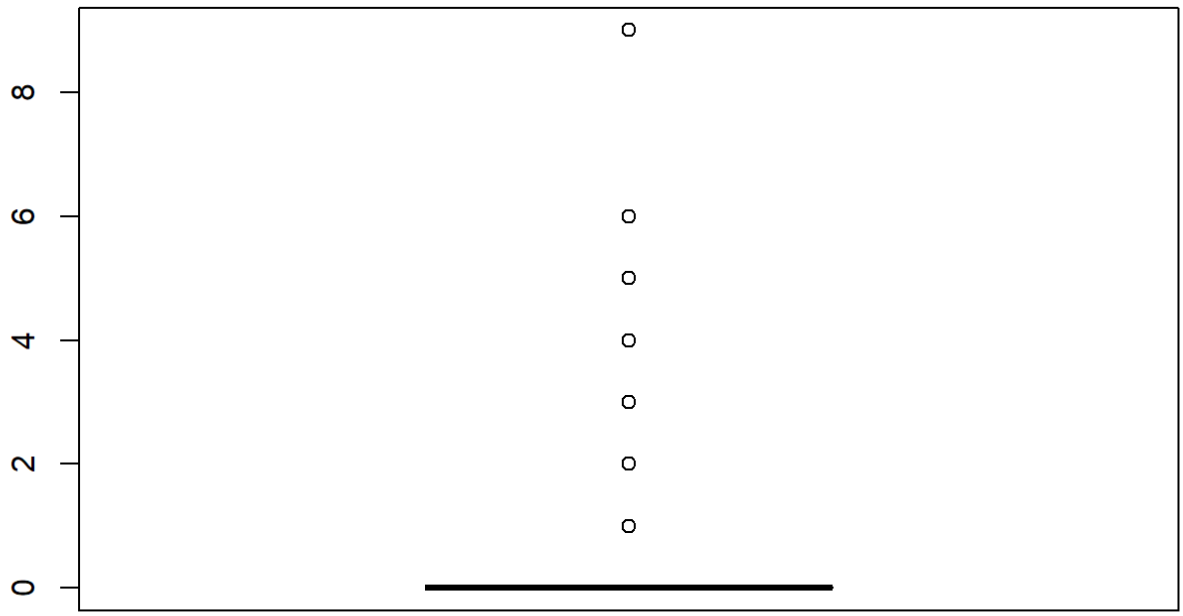
Al realizar el gráfico boxplot, se observan ciertos valores extremos a partir de 60 años y para valores cercanos a 0. Sabiendo que se trata de una columna que indica la edad de los pasajeros, consideramos que son datos reales y por lo tanto no habría que realizarles ningún tratamiento.

```
boxplot(datosTitanic$SibSp)
```



Por otro lado, observamos que para la variable SibSp aparecen cuatro registros extremos con valores 3, 4, 5 y 8. Aunque 8 es un número elevado teniendo en cuenta que esta variable contiene el número de hermanos y esposas a bordo del titanic, es posible que viajara una familia con 8 hijos y de ahí ese valor extremo. Por lo tanto, aunque sea un dato que destaque en la población, se decide no modificarlo ya que no sería raro que se tratara de un dato real sabiendo en la epoca en la que ocurrió el accidente.

```
boxplot(datosTitanic$Parch)
```



Por otro lado, en la variable Parch se observa que la gran mayoría de los pasajeros viajaban sin padres o hijos. Por otro lado, toma valores desde 1 hasta 9. Nueve podría considerarse un valor extremo, pero seguramente proceda del registro del padre de la familia anteriormente mencionada que viajara con sus ocho hijos más su padre.

De este modo, parece que aunque aparezcan valores extremos, son valores que son reales y por lo tanto no se realiza ningún tratamiento sobre ellos.

4 Análisis de los datos

Sabiendo que para tratar los datos es mejor hacer uso de números en vez de cadenas de texto, se decide transformar las variables categóricas en numéricas. De tal forma, que cada categoría se sustituirá por un valor numérico, este será el índice de la posición en la que se encuentran a continuación a la hora de especificarlas. Por ejemplo en el caso de embarked se ha especificado el orden “S”-“C”-“Q”-“Desconocido”, “S” tomará el valor 1, “C” el 2, “Q” el 3 y “Desconocido” el 4.

```
#Creamos los vectores que usaremos para la transformación
embarked <- c("S","C","Q", "Desconocido")
sex <- c("male","female")
alone <- c("yes","no")

#Inicializamos los datos
datosNorm = datosTitanic

#Realizamos la transformación
datosNorm$Embarked <- match(datosTitanic$Embarked, embarked)
```

```
datosNorm$Sex<- match(datosTitanic$Sex, sex)
datosNorm$alone <- match(datosTitanic$alone, alone)

#Mostramos la transformación realizada
head(datosNorm)
```

```
##   Survived Pclass Sex   Age SibSp Parch Embarked alone
## 1        0      3   1 22.00     1     0        1     2
## 2        1      1   2 38.00     1     0        2     2
## 3        1      3   2 26.00     0     0        1     1
## 4        1      1   2 35.00     1     0        1     2
## 5        0      3   1 35.00     0     0        1     1
## 6        0      3   1 22.75     0     0        3     1
```

En algunos algoritmos, el tener variables con distintas escalas hace que los pesos de las distintas variables sean distintos. Por lo tanto, se decide normalizar los datos para que las variables tomen valores entre 0 y 1.

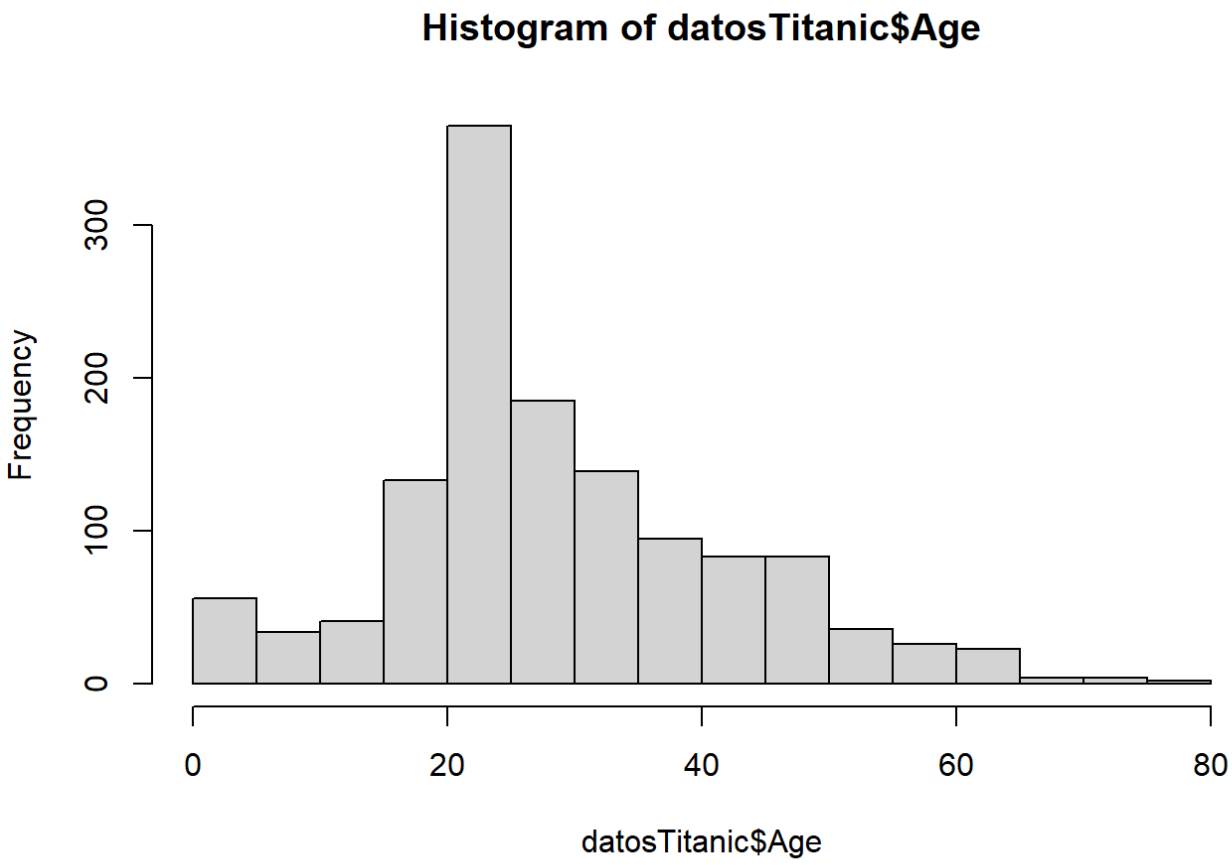
```
#Función normalizar
normalizar <- function(x)
{
  return ((x - min(x)) / (max(x) - min(x)))
}

#Se aplica la función
datosNorm$Age <- normalizar(datosNorm$Age)
datosNorm$SibSp <- normalizar(datosNorm$SibSp)
datosNorm$Parch <- normalizar(datosNorm$Parch)
```

4.1 Comprobación de la normalidad y homogeneidad de la varianza

Analicemos la distribución de la edad para comprobar si tiene una distribución normal, ya que esto sería de gran ayuda para realizar lo análisis. Mostramos el histograma de esta variable para comprobar la distribución que tienen los datos.

```
hist(datosTitanic$Age)
```



Se observa un pico de pasajeros entre las edades de 20 y 25 años. Al aplicar el text de shapiro wilk obtenemos un p valor menor de 0,05, por lo tanto se tiene que rechazar la hipótesis nula y no se puede afirmar que la variable Age siga una distribución normal.

```
library(nortest)
shapiro.test(datosTitanic$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datosTitanic$Age
## W = 0.96479, p-value < 2.2e-16
```

Aplicamos el text de **Kolmogorov-Smirnov** que es algo menos restrictivo, con el objetivo de comprobar si con este text se podría aceptar la hipotesis nula de normalidad. Pero al aplicar el text obtenemos un p valor muy pequeño, por lo tanto se sigue sin poder aceptar la normalidad.

```
ks.test(datosTitanic$Age, pnorm, mean(datosTitanic$Age), sd(datosTitanic$Ag
e))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  datosTitanic$Age
```

```
## D = 0.10392, p-value = 1.055e-12
## alternative hypothesis: two-sided
```

Al no poder aceptar la normalidad, realizamos la **transformación BoxCox** sobre la variable Age para conseguir la normalidad. Una vez aplicada la transformación volvemos a aplicar el text de Shapiro, pero seguimos obteniendo un valor muy pequeño. Al no conseguir tener una distribución normal en la variable Age se tendrá que tener en cuenta que solo se podrán aplicar text no paramétricos que tienen una menor potencia estadística.

```
library(DescTools)
shapiro.test(
  BoxCox(datosTitanic$Age, lambda = BoxCoxLambda(datosTitanic$Age))
)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BoxCox(datosTitanic$Age, lambda = BoxCoxLambda(datosTitanic$Age))
## W = 0.96463, p-value < 2.2e-16
```

Analicemos ahora si hay diferencias significativas en las distintas variables segun si los pasajeros sobreviven o no. Al aplicar el text **Fligner-Killeen** se obtiene un p valor mayor a 0,05. Por lo tanto se puede aceptar la hipótesis nula de que la varianza en la edad es idéntica en el grupo de pasajeros que sobreviven y los que no.

```
fligner.test( Age ~ Survived, data = datosNorm )
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 3.1204, df = 1, p-value = 0.07732
```

Se aplica el text **Mann-Whitney-Wilcoxon** no paramétrico, para comprobar si ambos grupos tienen la variable Age igualmente distribuida. Al aplicarlo se obtiene un p valor alto, por lo tanto se puede decir que ambos grupos tienen la misma distribución en la variable edad.

```
wilcox.test(Age ~ Survived, data = datosNorm )
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Age by Survived
## W = 203371, p-value = 0.7553
## alternative hypothesis: true location shift is not equal to 0
```

A continuación se aplica el text para comprobar la distribución de los dos grupos en las distintas variables.

```
#Pclass
chisq.test(table(datosNorm$Survived, datosNorm$Pclass))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(datosNorm$Survived, datosNorm$Pclass)
## X-squared = 91.724, df = 2, p-value < 2.2e-16
```

```
#Sex
chisq.test(table(datosNorm$Survived, datosNorm$Sex))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(datosNorm$Survived, datosNorm$Sex)
## X-squared = 617.31, df = 1, p-value < 2.2e-16
```

```
#Alone
chisq.test(table(datosNorm$Survived, datosNorm$alone))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(datosNorm$Survived, datosNorm$alone)
## X-squared = 60.333, df = 1, p-value = 8.008e-15
```

```
#Emabarked
chisq.test(table(datosNorm$Survived, datosNorm$Embarked))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(datosNorm$Survived, datosNorm$Embarked)
## X-squared = 27.964, df = 3, p-value = 3.695e-06
```

```
#SibSp
chisq.test(table(datosNorm$Survived, datosNorm$SibSp))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(datosNorm$Survived, datosNorm$SibSp)
## X-squared = 44.565, df = 6, p-value = 5.711e-08
```

```
#Parch
chisq.test(table(datosNorm$Survived, datosNorm$Parch))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(datosNorm$Survived, datosNorm$Parch)
## X-squared = 45.827, df = 7, p-value = 9.445e-08
```

En ningún caso se obtiene un p-valor superior a 0,05. Por lo tanto, se puede afirmar que en las variables Pclass, Sex, alone, Embarked, SibSp y Parch se observan diferencias estadisticamente significativas en los pasajeros que sobreviven y los que no. Por lo tanto, serán variables que nos ayudarán a la hora de clasificar los datos en dos grupos distintos.

4.2 Aplicación de pruebas de contraste de hipótesis, correlaciones, regresiones...

4.2.1 Correlaciones

Al aplicar el **test de correlación de Spearman**, se obtiene un p-valor superior a 0,05. Por lo tanto, las variables Survived y Age no están correlacionadas.

```
cor.test(datosNorm$Survived, datosNorm$Age, method="spearman")
```

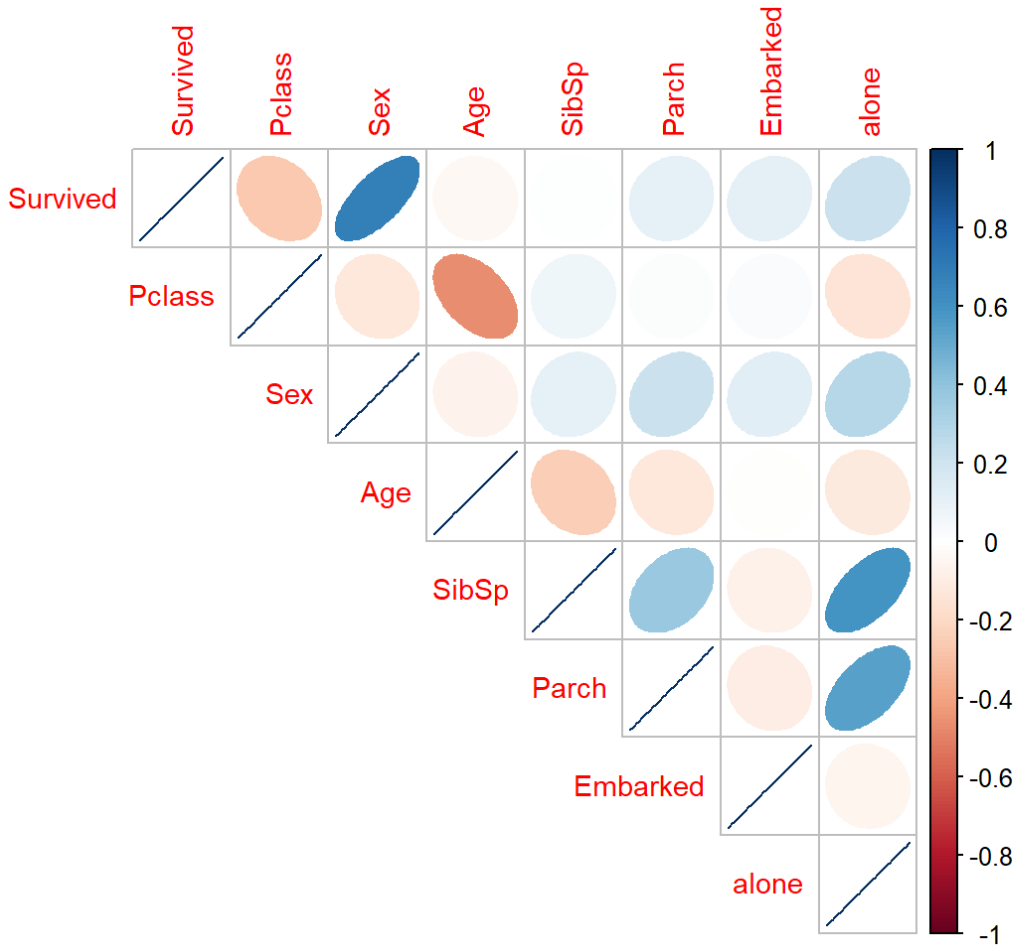
```
##
##  Spearman's rank correlation rho
##
## data:  datosNorm$Survived and datosNorm$Age
## S = 377046269, p-value = 0.7554
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.008619155
```

En este gráfico se puede observar mediante los colores la matriz de correlaciones por medio de la función *corrplot*. Este gráfico indica que entre más intensidad del color, ya sea azúl o rojo, mayor es la correlación, colores ténues significan correlación baja.

Al observar que variables están correlacionadas con la variable de supervivencia, se obtiene que hay una relación directamente proporcional con el sexo. Es decir, que si eres mujer tienes más probabilidades de sobrevivir.

Por otro lado, se observa una ligera correlación inversa entre la variable Pclass y la supervivencia, esto quiere decir que contra menor fuera la clase en la que viajabas, tienes menos posibilidades de sobrevivir.

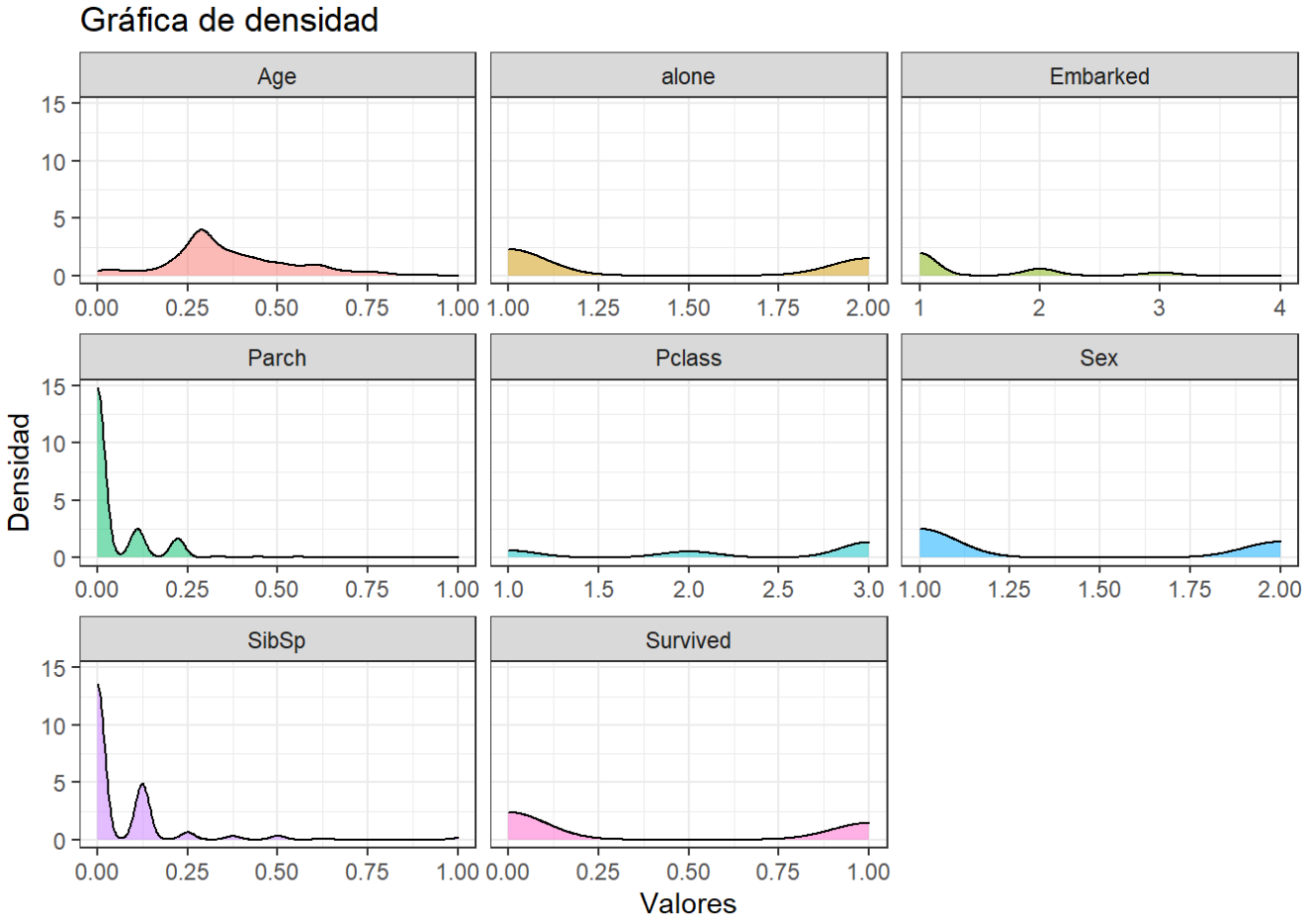
```
library(corrplot)
corrplot(cor(datosNorm), type="upper", method="ellipse", tl.cex=0.9)
```



A continuación se muestra una gráfica de densidad donde se puede observar cómo de distribuidos están los datos, en el caso de haber "picos" como en la edad, significa que muchos de los pasajeros se encuentran en ese rango, cuanto más plana sea la gráfica más uniformemente están distribuidos los registros.

```
library(tidyverse)

datosNorm %>%
  gather(Attributes, value, 1:8) %>%
  ggplot(aes(x=value, fill=Attributes)) +
  geom_density(colour="black", alpha=0.5, show.legend=FALSE) +
  facet_wrap(~Attributes, scales="free_x") +
  labs(x="Valores", y="Densidad",
       title="Gráfica de densidad") +
  theme_bw()
```



Una vez realizado el análisis exploratorio de los datos, empecemos a realizar los modelos predictivos. Antes de comenzar con esta parte, dividimos el conjunto de datos en datos de entrenamiento y datos de validación. Así se podrá comprobar la eficacia del algoritmo generado en datos no utilizados en la parte de entrenamiento, evitando de este modo la sobreexpecialización del modelo a los datos proporcionados.

```
library(rminer)
h <- holdout(datosNorm$Survived, ratio=2/3, mode="stratified")
titanic_train <- datosNorm[h$tr,]
titanic_test <- datosNorm[h$ts,]
```

Observamos que al realizar la división en datos de entrenamiento y validación, se obtiene una proporcion similar de personas que sobreviven en ambos conjuntos.

```
print(table(titanic_train$Survived))
```

```
##
##   0   1
## 550 322
```

```
print(table(titanic_test$Survived))
```

```
##
##   0   1
## 265 172
```

4.2.2 Regresión

Generamos un modelo de regresión logística para predecir los pasajeros que sobreviven y los que no. Se observa que el mayor peso lo proporciona la variable Sexo, que será la variable que más separe a los dos grupos.

```
modelo1 <- glm(Survived ~ Pclass+Sex+SibSp+Parch+Embarked+Age, data=titanic_
train, family="binomial")
summary(modelo1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + SibSp + Parch + Embarked +
##     Age, family = "binomial", data = titanic_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6840  -0.5298  -0.3785   0.4851   2.5525
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.56799    0.56086  -4.579 4.68e-06 ***
## Pclass      -1.02405    0.14206  -7.208 5.66e-13 ***
## Sex          3.69116    0.23010  16.042  < 2e-16 ***
## SibSp       -3.57136    0.99995  -3.572 0.000355 ***
## Parch       -0.07397    1.05704  -0.070 0.944207
## Embarked     0.31134    0.15841   1.965 0.049362 *
## Age         -2.81478    0.69179  -4.069 4.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1148.54  on 871  degrees of freedom
## Residual deviance:  648.67  on 865  degrees of freedom
## AIC: 662.67
##
## Number of Fisher Scoring iterations: 5
```

Una vez creado el modelo, predecimos los valores para los datos de test y mostramos la matriz de confusión. El modelo predice correctamente si el pasajero sobrevive o no en el 84% de las ocasiones.

```
pred_log <- predict(modelo1, newdata=titanic_test)

table(titanic_test$Survived, pred_log > 0.5)
```

```
##
##     FALSE TRUE
```

```
##    0   245    20
##    1    39   133
```

4.2.3 Random Forest

Veamos si podemos mejorar la predicción haciendo creando un modelo random forest de validación cruzada con 5 folds.

```
library(caret)
train_control <- trainControl(method="cv", number = 5)
modelo_rand_for <- train(Survived~., data = titanic_train, method="rf", trCo
ntrol = train_control)
```

Con el modelo creado predecimos los valores en los datos de test y mostramos la matriz de confusión. En este caso se obtiene una precisión del 85%, ligeramente superior al modelo anterior.

```
titanic_pred <- predict(modelo_rand_for, newdata=titanic_test)

table(titanic_test$Survived, titanic_pred > 0.5)
```

```
##
##      FALSE TRUE
##    0   244   21
##    1    36  136
```

5 Conclusiones

Este problema quería responder la cuestión de si un pasajero dadas sus características como edad o sexo consigue sobrevivir a la catástrofe del Titanic. Gracias a los análisis y los modelos generados que, con una precisión de casi un 85%, somos capaces de predecir dicha cuestión a partir de las variables proporcionadas.

Contribuciones	**Firma**
Investigación previa	Marta Rodríguez y Pedro Félez
Redacción de las respuestas	Marta Rodríguez y Pedro Félez
Desarrollo código	Marta Rodríguez y Pedro Félez