PRAC 2: Limpieza y validación de los datos

Javier Jiménez Reyes

22 de mayo de 2019

0. Tratamiento del dataframe: Titanic: Machine Learning from Disaster

A continuación se desarrollan las principales etapas de un proyecto analítico para el tratamiento del dataframe **Titanic: Machine Learning from Disaster** (https://www.kaggle.com/c/titanic). Cada una de la tareas a realizar se abordarán individualmente indicando los comandos en **R** utilizados en cada uno de los pasos.

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

La pregunta que se pretende resolver con el análisis del dataframe **Titanic** es averiguar qué tipo de personas tenían probabilidades de sobrevivir. Así pues el **objetivo** es: predecir qué pasajeros sobrevivieron a la tragedia).

Para poder describir el dataframe primero de todo se realizará la carga de los datos. La página web donde se encuentra el repositorio de datos, divide los registros en dos archivos descargables (**trian.csv** y **test.csv**):

```
##Carga archivo de train.csv
direccion<-getwd()
direccion<-paste(direccion,"/titanic/train.csv",sep="")
titanic_train<-read.csv(direccion,header=TRUE)

#Carga archivo de test.csv
direccion<-getwd()
direccion<-paste(direccion,"/titanic/test.csv",sep="")
titanic_test<-read.csv(direccion,header=TRUE)
```

Con los datos ya cargados se realiza un primer análisis de la configuración de las tablas y de que información contienen.

Nombre de las variables del dataframe **titanic_train**:

```
names(titanic_train)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"
```

Nombre de las variables del dataframe **titanic_test**:

```
names(titanic_test)
```

```
##  [1] "PassengerId" "Pclass"      "Name"        "Sex"         "Age"
##  [6] "SibSp"       "Parch"       "Ticket"      "Fare"        "Cabin"
## [11] "Embarked"
```

Tal y como se puede observar el dataframe **titanic_train** se compone de una variable más que el dataframe **titanic_test**. Esta variable extra corresponde al atributo **Survived**, que indica si el tripulante registrado sobrevivió (valor 1) o no (valor 0).

De la web **Titanic: Machine Learning from Disaster** (https://www.kaggle.com/c/titanic/data) desde donde se ha descargado el repositorio de datos, se obtiene el siguiente diccionario de variables:

Variable	Descripción	Valores
survival	Superviviente	0=No, 1=Si
pclass	Clase billete	1=1st,2=2nd, 3=3rd
sex	Sexo	male/female
Age	Edad (en años)	
sibsp	Nº de hermanos/conyugues a bordo	
parch	Nº de padres/hijos a bordo	
ticket	Nº de ticket	
fare	Tarifa de pasajero	
cabin	Nº del camarote	
embarked	Puerto de embarque	C=Cherbourg,Q=Queenstown,S=Southampton

Se analiza ahora el número de registros de cada dataframe:

```
#Nº de observaciones dataframe titanic_train:
cat("El número de registros del dataframe titanic_trian es: ", nrow(titanic_train))
```

```
## El número de registros del dataframe titanic_trian es:  891
```

```
#Nº de observaciones dataframe titanic_test:
cat("El número de registros del dataframe titanic_test es: ", nrow(titanic_test))
```

```
## El número de registros del dataframe titanic_test es:  418
```

Tal y como se puede observar el nº de registros del dataframe **titanic_train** es algo mayor que el doble de observaciones del dataframe **titanic_test**.

Una vez analizada la composición de cada uno de los dataframe, se analizan si el tipo de objeto que lo representa es el correcto. Para verificar el tipo de variable que R ha asignado a cada uno de los atributo al realizar la lectura y carga de los archivos.

Dataframe **titanic_test**:

```
res <- sapply(titanic_test,class)
knitr::kable(data.frame(Variables=names(res),Clase=as.vector(res)))
```

Variables	Clase
PassengerId	integer
Pclass	integer
Name	factor
Sex	factor
Age	numeric
SibSp	integer
Parch	integer
Ticket	factor
Fare	numeric
Cabin	factor
Embarked	factor

Dataframe **titanic_train**:

```
res <- sapply(titanic_train,class)
knitr::kable(data.frame(Variables=names(res),Clase=as.vector(res)))
```

Variables	Clase
PassengerId	integer
Survived	integer
Pclass	integer
Name	factor
Sex	factor
Age	numeric
SibSp	integer
Parch	integer
Ticket	factor
Fare	numeric
Cabin	factor
Embarked	factor

Como se puede observar en los resultados es necesario de ajustar el tipo de variable de algunos de los resultados. Las variables **Survived** y **Pclass** han de convertirse a **factores** puesto que pese a mostrar valores numéricos, realmente el tipo de dato que hay tras este valor es cualitativo y no cuantitativo.

Si ahora se analiza la clase de las variables se obtiene:

```
#Clase variable Survived
class(titanic_train$Survived)
```

```
## [1] "factor"
```

```
#Calse variable Pclass
class(titanic_train$Pclass)
```

```
## [1] "factor"
```

```
class(titanic_test$Pclass)
```

```
## [1] "factor"
```

2. Integración y selección de los datos de interés a analizar.

En el siguiente apartado se integran todos los datos en una sola tabla para su posterior análisis. Para poder integrar la totalidad de los datos en una sola tabla que permita su posterior tratamiento será necesario igualar el número de variables. Tal y como se pudo ver en el anterior apartado el dataframe **titanic_train** tiene el atributo **Survived** por lo que se crea el mismo atributo en el dataframe **titanic_test** y se rellena con valores nulos (NA).

```
titanic_test$Survived <- NA
```

Notar que el motivo de realizar la transformación de la variable **Survived** a factor únicamente en el dataframe titanic_train y no en titanic_test, es debido a que no interesa contemplar los valores NA como factores, sino que se sigan interpretando como **valores nulos**.

Una vez los dos dataframe tienen la misma estructura se genera un nuevo atributo en cada uno de ellos para poderlos diferenciar y poderlos volver a separar en los dos dataframe iniciales (test y train), una vez depurada la información. Para ello se crea el atributo **TrainSet**.

En el caso del dataframe **titanic_train** este atributo tomará valores verdaderos para cada registro, mientras que para el caso del dataframe **titanic_test**, tomara valores falsos.

```
#Creación del atributo TrainSet en los dos dataframe
titanic_train$TrainSet<-TRUE
titanic_test$TrainSet<-FALSE
```

Una vez los dos data set ya están preparados, se procede a su integración en una única tabla/repositorio de datos.

```
titanic_completo<-rbind(titanic_train,titanic_test)
```

Si se analiza ahora la tabla resultante se observa que:

```
#Nº de registros de la tabla titanic_completo
cat("El nº de registros es: ",nrow(titanic_completo))
```

```
## El nº de registros es:  1309
```

```
#Nº de columnas de la tabla titanic_completo
cat("El nº de columnas es: ",ncol(titanic_completo))
```

```
## El nº de columnas es:  13
```

Por ahora en este apartado se decide no eliminar ningún campo por si puede ser de utilidad a la hora de realizar la limpieza de datos, relacionando conceptos que permitan rellenar los registros vacíos o nulos. La decisión de que atributos eliminar o dejar fuera para la elaboración del modelo predictivo que permita dar respuesta al caso de estudio, se realizará al inicio del apartado 4.

3. Limpieza de los datos.

En el siguiente apartado se procede a detectar la existencia ceros o elementos vacíos y a su gestión.

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

Para la detección de valores vacíos y nulos se utiliza la siguiente función:

```
summary(titanic_completo)
```

```
##   PassengerId   Survived   Pclass                                  Name
##  Min.   :   1   0   :549   1:323   Connolly, Miss. Kate              :   2
##  1st Qu.: 328   1   :342   2:277   Kelly, Mr. James                  :   2
##  Median : 655   NA's:418   3:709   Abbing, Mr. Anthony               :   1
##  Mean   : 655                      Abbott, Mr. Rossmore Edward       :   1
##  3rd Qu.: 982                      Abbott, Mrs. Stanton (Rosa Hunt):   1
##  Max.   :1309                      Abelson, Mr. Samuel               :   1
##                                    (Other)                           :1301
##      Sex           Age            SibSp            Parch
##  female:466   Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  male  :843   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##               Median :28.00   Median :0.0000   Median :0.000
##               Mean   :29.88   Mean   :0.4989   Mean   :0.385
##               3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##               Max.   :80.00   Max.   :8.0000   Max.   :9.000
##               NA's   :263
##       Ticket          Fare                      Cabin      Embarked
##  CA. 2343:  11   Min.   :  0.000                  :1014    :  2

##  1601    :   8   1st Qu.:  7.896   C23 C25 C27    :   6   C:270
##  CA 2144 :   8   Median : 14.454   B57 B59 B63 B66:   5   Q:123
##  3101295 :   7   Mean   : 33.295   G6             :   5   S:914
##  347077  :   7   3rd Qu.: 31.275   B96 B98        :   4
##  347082  :   7   Max.   :512.329   C22 C26        :   4
##  (Other) :1261   NA's   :1         (Other)        : 271
##   TrainSet
##  Mode :logical
##  FALSE:418
##  TRUE :891
##
##
##
##
```

Tal y como se puede observar las variables que contienen algún valor nulo (NA) o vacío ( ) son las siguientes:

- El atributo **Survived** contiene 418 registros nulos (NA). Estos corresponden a los 418 registros del dataframe **titanic_test**. Estos se completarán con el modelo predictivo que se defina.
- El atributo **Age** contiene 263 registros nulos (NA).
- El atributo **Fare** contiene 1 registro nulo (NA).
- El atributo **Cabin** contiene 1014 registros vacíos ( ).
- El atributo **Embarked** contiene 2 registros vacíos ( ).

A continuación se irán tratando individualmente cada uno de los casos detectados para los atributos: **Fare**, **Cabin**, **Embarked** y **Age**.

Gestión del atributo Fare:

Para calcular el valor por el que substituir el valor nulo del atributo **Fare**, se decide utilizar en este caso el método de calcular **la media del atributo para las muestra de la misma clase**. Para ello el atributo que se emplea como elemento clasificador, es el atributo **Pclass**.

```
#Cálculo de los valores medios
res<-tapply(titanic_completo$Fare,titanic_completo$Pclass, mean, na.rm=TRUE)
#Se muestra el resultado
res
```

```
##        1        2        3
## 87.50899 21.17920 13.30289
```

Una vez obtenidos los valores promedio del atributo **Fare** según la clase del pasajero, es necesario conocer la cual es la clase del registro con el valor nulo.

```
#Obtención del registro con valor nulo para el atributo Fare
res<-titanic_completo[is.na(titanic_completo$Fare),]
#Clase del tripulante del registro con el valor nulo
res$Pclass
```

```
## [1] 3
## Levels: 1 2 3
```

Se obtiene que el tripulante con el valor de **Fare** nulo pertenece a la **Pclass** 3 que ha dado un valor promedio de 13.303.

Con el objetivo de observar variaciones en el resultado en el caso de añadir más atributos clasificadores en el cálculo del valor promedio, se añade el atributo **Sex** al cálculo.

```
#Cálculo de los valores medios
res<-tapply(titanic_completo$Fare,  list(titanic_completo$Sex,titanic_completo$Pclass),  mean,
 na.rm=TRUE)
#Se muestra el resultado
res
```

```
##                1        2        3
## female 109.41238 23.23483 15.32425
## male    69.88838 19.90495 12.41546
```

Para el registro con el valor **Fare** nulo el valor de **Sex** es:

```
#Obtención del registro con valor nulo para el atributo Fare
res<-titanic_completo[is.na(titanic_completo$Fare),]
#Clase del tripulante del registro con el valor nulo
res$Sex
```

```
## [1] male
## Levels: female male
```

Así pues si el valor promedio de los registro con valor **Sex = male** y **Pclass = 3** es de 12.415. Se puede observar que la diferencia entre los dos valores calculados es pequeña y al tratarse únicamente de un valor que reemplazar en una tabla de 1309 registros, esta variación no es significativa. Por lo tanto se decide substituir el valor nulo para el atributo **Fare** por el valor de **13.30289**.

```
titanic_completo[is.na(titanic_completo$Fare),"Fare"]<-13.30289
```

Gestión del atributo Cabin:

En el caso del atributo **Cabin** debido a que el nº de registros vacíos es del 70% de los registros, se decide eliminar esta columna del dataframe.

```
titanic_completo$Cabin<-NULL
```

Gestión del atributo Embarked:

En el caso del atributo **Embarked** al no tratarse de un valor numérico como los atributos **Age** y **Fare**, no se puede calcular un valor promedio como tal. Por este motivo el método para decidir con qué valor cumplimentar los dos valores vacíos es observando la correlación entre los atributos **Embarked** y **Pclass**.

```
table(titanic_completo$Pclass,titanic_completo$Embarked)
```

```
##
##           C   Q   S
##   1   2 141   3 177
##   2   0  28   7 242
##   3   0 101 113 495
```

Tal y como se puede observar en el resultado de la tabla obtenida los dos valores vacíos corresponden a registros de dos tripulantes que viajan en **primera clase** (Pclass = 1). Si se observa cómo se distribuyen los registros que pertenece a esta clase turística, se puede apreciar que el embarque se hace principalmente entre la puerta **C** y **S**, siendo esta última la que tiene más registros. De igual modo para el resto de pasajeros de 2a y 3a clase, el embarque principalmente se realiza por la puerta **S**.

Notar que si nuevamente aparte de tener en cuenta el atributo **Pclass** se añadiese también el atributo **Sex** el resultado sería el siguiente:

```
table(titanic_completo$Pclass,titanic_completo$Embarked,titanic_completo$Sex)
```

```
## , ,  = female
##
##
##           C   Q   S
##   1   2  71   2  69
##   2   0  11   2  93
##   3   0  31  56 129
##
## , ,  = male
##
##
##           C   Q   S
##   1   0  70   1 108
##   2   0  17   5 149
##   3   0  70  57 366
```

En este caso se puede observar como el valor mayoritario de puerta de embarque al que pertenecen los registros del mismo **Sex** (femenino) y **Class** (primera clase) que el de los dos registros con valores de embarque vacíos, es la puerta de embarque **C**. Así pues en este caso el resultado variaría. No obstante tal y como se comentó en la depuración de la variable **Age**, lo adecuado sería mirar si seleccionando la **C** como resultado adecuado para substituir el valor vacío, el modelo final se ajusta mejor.

Por lo tanto se decide substituir para los dos registros con valor vacío para el atributo **Embarked** por el valor **S**.

```
titanic_completo[titanic_completo$Embarked=='',"Embarked"]<-'S'
```

Gestión del atributo Age:

En el caso del atributo **Age** se puede observar que son varios los registros que contienen un valor nulo. Así pues intentar completarlos analizándolos a título individual cada uno de ellos sería una tarea compleja. Por ese motivo las opciones propuestas dos:

- Medainte el uso de la **media del atributo para las muestra de la misma clase**.
- Mediante el **valor más probable** por medio de técnicas como la regresión.

Para este caso se decide escoger la opción del **valor más probable** por medio de técnicas como la regresión como método de resolución. Lo adecuado e interesante sería una vez resuelto el ejercicio y obtenidos los resultados, ver si depurando este atributo por medio de la primera opción el ajuste del modelo predictivo sería igual de bueno o no. Con este apunte se pretende remarcar una de las características importantes en los proyectos analíticos de ciencia de datos: el proceso de un proyecto de ciencia de datos es **iterativo**.

Lo primero que se realizará será crear el modelo de regresión formado por los atributos **Pclass**, **SibSp**, **Parch** y **Embarked**:

```
#Definición de la equación del modelo de regresión
ecuacion="Age ~ Pclass + SibSp + Parch + Embarked"
#Cálculo del modelo de regresión
regresion<-lm(formula = ecuacion, data=titanic_completo)
```

Una vez generado el modelo se aplica para calcular el valor del atributo **Age** para los registros con valor nulo. Para ello es necesario crear un *sub dataframe* únicamente con los registros con valor del atributo **Age** nulos y con los atributos utilizados en la ecuación de la regresión (**Pclass**, **SibSp**, **Parch** y **Embarked**):

```
#Creación del sub dataframe
Age.NA<-titanic_completo[is.na(titanic_completo$Age),c("Pclass","SibSp","Parch","Embarked")]
#Aplicación de la regresión para el cálculo de los valores nulos de Age
prediccion.Age<-predict(regresion,newdata = Age.NA)
```

Una vez calculados los valores se procede a su substitución en el dataframe original:

```
titanic_completo[is.na(titanic_completo$Age),"Age"]<-prediccion.Age
```

Tras realizar las distintas acciones para tratar los registros con valores nulos y vacíos se vuelve a verificar la situación de los distintos atributos.

```
summary(titanic_completo)
```

```
##   PassengerId   Survived   Pclass                                  Name
##  Min.   :   1   0   :549   1:323   Connolly, Miss. Kate              :   2
##  1st Qu.: 328   1   :342   2:277   Kelly, Mr. James                  :   2
##  Median : 655   NA's:418   3:709   Abbing, Mr. Anthony               :   1
##  Mean   : 655                      Abbott, Mr. Rossmore Edward       :   1
##  3rd Qu.: 982                      Abbott, Mrs. Stanton (Rosa Hunt):   1
##  Max.   :1309                      Abelson, Mr. Samuel               :   1
##                                    (Other)                           :1301
##      Sex           Age             SibSp            Parch
##  female:466   Min.   :-1.134   Min.   :0.0000   Min.   :0.000
##  male  :843   1st Qu.:22.000   1st Qu.:0.0000   1st Qu.:0.000
##               Median :27.319   Median :0.0000   Median :0.000
##               Mean   :29.483   Mean   :0.4989   Mean   :0.385
##               3rd Qu.:36.408   3rd Qu.:1.0000   3rd Qu.:0.000
##               Max.   :80.000   Max.   :8.0000   Max.   :9.000
##
##       Ticket          Fare         Embarked  TrainSet
##  CA. 2343:  11   Min.   :  0.000    :  0    Mode :logical

##  1601    :   8   1st Qu.:  7.896   C:270    FALSE:418
##  CA 2144 :   8   Median : 14.454   Q:123    TRUE :891
##  3101295 :   7   Mean   : 33.280   S:916
##  347077  :   7   3rd Qu.: 31.275
##  347082  :   7   Max.   :512.329
##  (Other) :1261
```

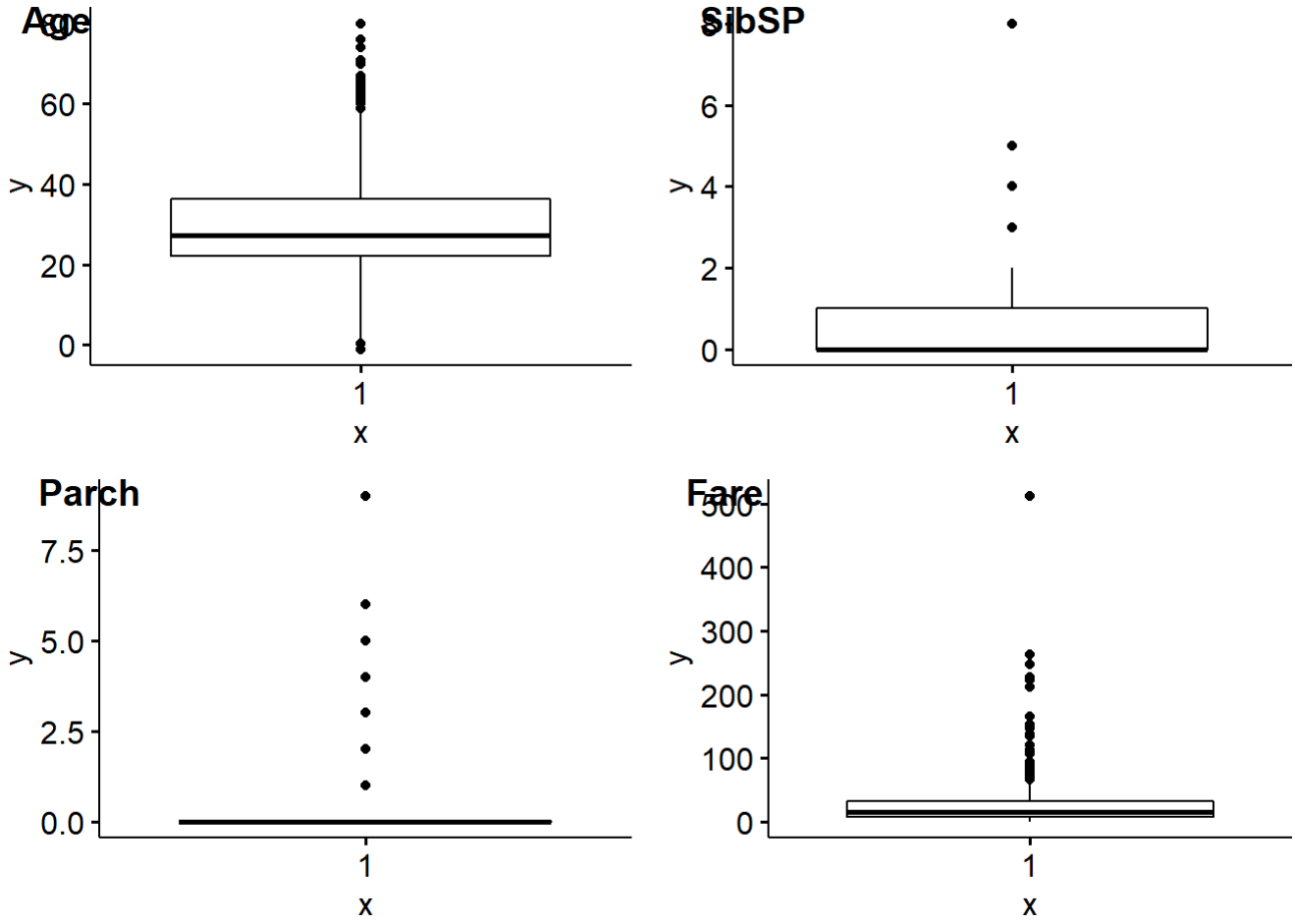
Como se puede observar todos los valores nulos o vacíos han sido tratados y ya no aparecen.

3.2. Identificación y tratamiento de valores extremos.

Para la detección de valores anomalos o extremos en las variables numéricas (**Age**, **SibSP**, **Parch** y **Fare**) se utiliza la representación gráfica de los valores estos atributos mediante un boxplot

```
bx.Age<-ggpubr::ggboxplot(titanic_completo$Age)
bx.SibSP<-ggpubr::ggboxplot(titanic_completo$SibSp)
bx.Parch<-ggpubr::ggboxplot(titanic_completo$Parch)
bx.Fare<-ggpubr::ggboxplot(titanic_completo$Fare)

ggpubr::ggarrange(bx.Age,bx.SibSP,bx.Parch,bx.Fare,labels=c("Age","SibSP","Parch","Fare"),nco
l = 2, nrow = 2)
```



Tal y como se puede apreciar en los gráficos en todos los atributos analizados, aparecen valores extremos.

Una vez detectados visualmente los valores extremos se buscan los bigotes del del boxplot para conocer los límites a partir de los que aparecen estos valores en el boxplot

```
bx.Age<-boxplot.stats(titanic_completo$Age)
bx.Age$stats
```

```
## [1]  0.42000 22.00000 27.31898 36.40773 58.00000
```

```
bx.SibSP<-boxplot.stats(titanic_completo$SibSp)
bx.SibSP$stats
```

```
## [1] 0 0 0 1 2
```

```
bx.Parch<-boxplot.stats(titanic_completo$Parch)
bx.Parch$stats
```

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

```
bx.Fare<-boxplot.stats(titanic_completo$Fare)
bx.Fare$stats
```

```
## [1]  0.0000  7.8958 14.4542 31.2750 65.0000
```

Para cumplimentar esta información se extraen por cada una de las variables el nº de valores extremos:

```
#Nº de valores extremos
#Age
length(boxplot(titanic_completo$Age,plot=FALSE)$out)
```

```
## [1] 53
```

```
#SibSP
length(boxplot(titanic_completo$SibSp,plot=FALSE)$out)
```

```
## [1] 57
```

```
#Parch
length(boxplot(titanic_completo$Parch,plot=FALSE)$out)
```

```
## [1] 307
```

```
#Fare
length(boxplot(titanic_completo$Fare,plot=FALSE)$out)
```

```
## [1] 171
```

En base al contaje de valores extremos detectados se podría aplicar como tratamiento para las variables **Age** y **SibSp**, eliminar la totalidad del registro por no tratarse de un número significativo de registros, mientras que en el caso de las variables **Parch** y **Fare** se podría aplicar una substitución de los valores extremos por el valor promedio o valor del bigote superior o inferior (según el caso) calculado anteriormente, ya que su volumen es mayor. No obstante se decide aplicar la segunda opción de tratamiento (substitución por el valor) en todos los atributos.

Partiendo de esta premisa se continúa analizando para valorar si se aplica dicho criterio a todos los valores extremos o únicamente a unos cuantos.

Se analiza la distribución de estos valores extremos en las variables **SibSp** y **Parch**:

Atributo **SibSp**:

```
table(boxplot(titanic_completo$SibSp,plot=FALSE)$out)
```

```
##
##  3  4  5  8
## 20 22  6  9
```

Atributo **Parch**:

```
table(boxplot(titanic_completo$Parch,plot=FALSE)$out)
```

```
##
##   1   2   3   4   5   6   9
## 170 113   8   6   6   2   2
```

Partiendo de la información visual y numérica se toman las siguientes deciciones:

- En el caso de la variable **SibSp** aparece un valor extremo muy por encima del resto. Este corresponde a 9 registros que indican que el número de hermanos o esposas del tripulante es 8. Tanto los registros con un valor de **SibSp** de 8 como de 5 se asumen como valores anómalos dentro de los valores extremos y son los que se deciden tratar, mientras que los registros con un valor por debajo de 5 se asumen como correcto. Como acción se substituye el valor reduciendo a un valor de 4 (extremo que se ha considerado como asumible).

```
#Cambio de los valores extremos
titanic_completo$SibSp[titanic_completo$SibSp>4]=4
#Verificación
table(boxplot(titanic_completo$SibSp,plot=FALSE)$out)
```

```
##
##  3  4
## 20 37
```

- En el caso de la variable **Parch** referente al número de padres o niños abordo por el pasajero se decide tomar como valor extremo aceptable el valor 2. Por lo tanto se decide reducir todos los registros con un valor superior a 2 a éste.

```
#Cambio de los valores extremos
titanic_completo$Parch[titanic_completo$Parch>2]=2
#Verificación
table(boxplot(titanic_completo$Parch,plot=FALSE)$out)
```

```
##
##   1   2
## 170 137
```

- En cuanto a la variable **Fare** se decide subsitituïr la totalidad de los valores extremos por el valor del bigote superior del boxplot que corresponde a 65.

```
#Cambio de los valores extremos
titanic_completo$Fare[titanic_completo$Parch>bx.Fare$stats[5]]=bx.Fare$stats[5]
```

- Finalmente se gestiona la variable **Age** para la que se miran los distintos registros extremos existentes.

```
table(boxplot(titanic_completo$Age,plot=FALSE)$out)
```

```
##
## -1.13357057666658             0.17             0.33               59
##                 8                1                1                3
##                60             60.5               61               62
##                 7                1                5                5
##                63               64               65               66
##                 4                5                3                1
##                67               70             70.5               71
##                 1                2                1                2
##                74               76               80
##                 1                1                1
```

Como se puede observar aparecen valores negativos lo que no tiene sentido. Como medida correctiva se decide substituir los valores extremos inferiores por el valor del bigote inferior (0.42000) y en el caso de los valores extremos superiores se deciden dejar.

```
#Cambio de los valores extremos
titanic_completo$Age[titanic_completo$Age<bx.Age$stats[1]]=bx.Age$stats[1]
```

4. Análisis de los datos.

En este apartado se realizará tanto la selección campos que se utilizar para la generación del modelo de predicción y que se indicó en el apartado 2 que se realizaría en este punto del análisis, como la selección de los registros que se emplearán para el análisis.

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

Lo primero que se realizará será la selección de los campos que se deciden emplear para la generación del modelo predictivo, los cuales son:

- PassangerId. Se utilizará únicamente como índice pero no se empleará para la generación del modelo de predicción.
- Survived.
- Pclass.
- Sex.
- Age.
- SibSp.
- Parch.
- Fare.
- Embarked.
- TrainSet. Este último campo se empleará únicamente para realizar la separación de registros nuevamente en los grupos de datos de **train** y **test**.

Para mantener la tabla original se decide guardar estos nuevos campos en un nuevo dataframe

```
titanic2<-titanic_completo[,c(1,2,3,5,6,7,8,10,11,12)]
```

Una vez seleccionados los atributos necesarios para la generación del modelo de predicción y con los datos ya tratados se vuelven a generar los grupos de datos **train** que se utilizarán para generar el modelo, y el grupo **test** que permitirán probar el modelo generado. Se guardaran en dos nuevos dataframe para así guardar los archivos originales.

```
titanic_train2<-titanic2[titanic2$TrainSet==TRUE,]
titanic_test2<-titanic2[titanic2$TrainSet==FALSE,]
```

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

A continuación se verifica la normalidad de las variables numéricas:

```
shapiro.test(titanic_train2$Age)$p.value
```

```
## [1] 3.569947e-11
```

```
shapiro.test(titanic_train2$Fare)$p.value
```

```
## [1] 1.084045e-43
```

```
shapiro.test(titanic_train2$Parch)$p.value
```

```
## [1] 1.279016e-42
```

```
shapiro.test(titanic_train2$SibSp)$p.value
```

```
## [1] 7.088475e-42
```

Como se puede observar en todos los casos se retorna un p-valor muy inferior al 5% por lo que se rechaza la hipótesis nula (H0) que la **muestra estudiada proviene de una población con una distribución normal**.

Se calcula ahora la varianza y desviación estándar para las variables numéricas **Age** y **Fare**.

Variable **Age**:

```
mean(titanic_train2$Age)
```

```
## [1] 29.38216
```

```
var(titanic_train2$Age)
```

```
## [1] 182.8279
```

```
sd(titanic_train2$Age)
```

```
## [1] 13.52139
```

Se puede apreciar cómo respecto al valor promedio de edad de la muestra el valor se desvía en un +-13.5 años.

Variable **Fare**:

```
mean(titanic_train2$Fare)
```

```
## [1] 32.20421
```

```
var(titanic_train2$Fare)
```

```
## [1] 2469.437
```

```
sd(titanic_train2$Fare)
```

```
## [1] 49.69343
```

Para el caso de la variable Fare la desviación respecto al valor medio de 32.2 es de +-49.7.

4.3. 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.

Para la realización del modelo predictivo que permita predecir si el pasajero sobrevive o no se decide emplear un **modelo de regresión logístico**. Se realizarán varios modelos variando el número de variables que interviene, y de todos ellos se comprobarán los valores AIC (es una medida de calidad relativa de un modelo estadístico, la cual basa su decisión en la bondad de ajuste del modelo y la complejidad del modelo). Esta medida indica que cuanto más pequeño es AIC mejor es el modelo.

Generación de los modelos:

```
#MODELO 1
glm1<-glm(formula=titanic_train2$Survived~Pclass,data=titanic_train2,family="binomial")
#MODELO 2
glm2<-glm(formula=titanic_train2$Survived~Pclass+Sex,data=titanic_train2,family="binomial")
#MODELO 3
glm3<-glm(formula=titanic_train2$Survived~Pclass+Sex+Age,data=titanic_train2,family="binomia
l")
#MODELO 4
glm4<-glm(formula=titanic_train2$Survived~Pclass+Sex+Age+SibSp,data=titanic_train2,family="bi
nomial")
#MODELO 5
glm5<-glm(formula=titanic_train2$Survived~Pclass+Sex+Age+SibSp+Parch,data=titanic_train2,fami
ly="binomial")
#MODELO 6
glm6<-glm(formula=titanic_train2$Survived~Pclass+Sex+Age+SibSp+Parch+Fare,data=titanic_train
2,family="binomial")
#MODELO 7
glm7<-glm(formula=titanic_train2$Survived~Pclass+Sex+Age+SibSp+Parch+Fare+Embarked,data=titan
ic_train2,family="binomial")
```

Valores medida AIC:

```
#MODELO 1
AIC(glm1)
```

```
## [1] 1089.108
```

```
#MODELO 2
AIC(glm2)
```

```
## [1] 834.8884
```

```
#MODELO 3
AIC(glm3)
```

```
## [1] 817.4629
```

```
#MODELO 4
AIC(glm4)
```

```
## [1] 803.073
```

```
#MODELO 5
AIC(glm5)
```

```
## [1] 805.0147
```

```
#MODELO 6
AIC(glm6)
```

```
## [1] 806.1427
```

```
#MODELO 7
AIC(glm7)
```

```
## [1] 806.0776
```

Tal y como se puede observar en los resultados el modelo que retorna mejores resultados es el 4 modelo en el que se tienen en cuenta únicamente los atributos: **Pclass**, **Sex**, **Age** y **SibSp**.

A continuación y utilizando el **cuarto caso** como modelo para realizar la predicción de la supervivencia del pasajero, se procede a su análisis.

```
summary(glm4)
```

```
##
## Call:
## glm(formula = titanic_train2$Survived ~ Pclass + Sex + Age +
##     SibSp, family = "binomial", data = titanic_train2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7416  -0.5921  -0.4038   0.6122   2.4455
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.257839   0.433054   9.832  < 2e-16 ***
## Pclass2     -1.295672   0.268691  -4.822 1.42e-06 ***
## Pclass3     -2.525507   0.258111  -9.785  < 2e-16 ***
## Sexmale     -2.737247   0.195435 -14.006  < 2e-16 ***
## Age         -0.042970   0.008177  -5.255 1.48e-07 ***
## SibSp       -0.437352   0.112789  -3.878 0.000105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  791.07  on 885  degrees of freedom
## AIC: 803.07
##
## Number of Fisher Scoring iterations: 5
```

Partiendo de los resultados obtenidos se realizan las siguientes observaciones:

- Se puede apreciar que ninguna de las variables tiene un p-valor superior al 5%, por lo que en todos los casos se desestima la hipótesis nula (H0), implicando que el dichas variables por si solas no son significativas para el resultado del modelo.
- Se puede observar como el hecho de que el tripulante del registro sea hombre o pertenezca a la 2a o 3a clase, contribuye negativamente al modelo, haciendo que el resultado tienda a 0. Podemos ver en ello cierta lógica ya que en dicho suceso (el naufragio del Titanic), se conoce que la mayor parte de la gente que logró sobrevivir pertenecía a primera clase y se priorizaba el acceso a botes salvavidas primero a mujeres y niños.

A continuación se calcula la calidad del ajuste mediante la **matriz de confusión** (se supone un umbral de discriminación del 75%).

```
#Cálculo de los valores resultados con el modelo de regresión:
p<-predict(object =glm4,newdata = titanic_train2,type="response")

#Aplicación del umbral de discriminación del 75% a los resultados del modelo:
prediocciones <- ifelse(p > 0.75, 1, 0)

# Transformar en data.frame la lista de predicciones del modelo de regresión
lista<- data.frame(matrix(unlist(prediocciones), nrow=891, byrow=T),stringsAsFactors=FALSE)
names(lista)<-c("Survived")

# Crear tabla únicamente con los datos del campo Survived de la tabla original de datos
datos<-data.frame(titanic_train2$Survived)
names(datos)<-c("Survived")

# Conversión a factores la tabla datos
datos$Survived<-factor(datos$Survived)

# Conversión a factores los datos de la tabla de predicciones y escalado con los niveles de l
a tabla de datos.
lista$Survived<-factor(lista$Survived,levels = levels(datos$Survived))

#Matriz de confusión con los valores en factor
caret::confusionMatrix(lista$Survived,datos$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 539 187
##          1  10 155
##
##                Accuracy : 0.7789
##                  95% CI : (0.7502, 0.8058)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.482
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9818
##             Specificity : 0.4532
##          Pos Pred Value : 0.7424
##          Neg Pred Value : 0.9394
##              Prevalence : 0.6162
##          Detection Rate : 0.6049
##    Detection Prevalence : 0.8148
##       Balanced Accuracy : 0.7175
##
##        'Positive' Class : 0
##
```

Del análisis realizado se obtienen los siguientes resultados:

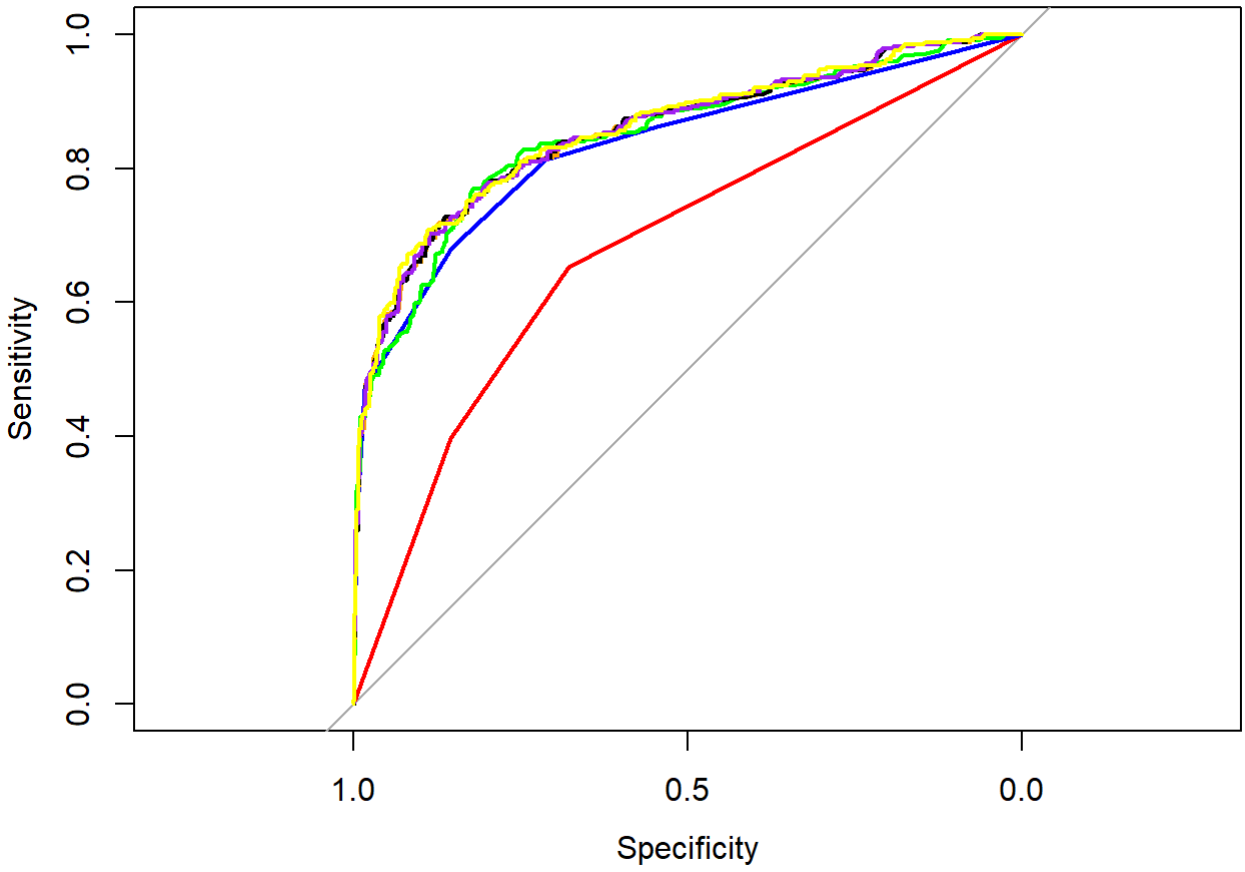
- El modelo predictivo ha calculado 726 registros con un valor Survived 0 (no ha sobrevivido) y 165 registros con un valor Survived 1 (ha sobrevivido).

- Los valores reales de la tabla son 549 registros con un valor Survived 0 (no ha sobrevivido) y 342 registros con un valor Survived 1 (ha sobrevivido).
- El total de **verdaderos positivos** es de: 155 registros donde la predicción ha indicado la supervivencia del tripulante y realmente la hay.
- El total de **verdaderos negativos** es de: 539 registros donde la predicción ha indicado la no supervivencia del tripulante y realmente no la hay.
- El total de **falsos positivos** es de: 10 registros donde la predicción ha indicado la supervivencia del tripulante y realmente no la hay.
- El total de **falsos negativos** es de: 187 registros donde la predicción ha indicado la no supervivencia del tripulante y realmente la hay.

Finalmente se emplea la curva ROC como otro método distinto para representar la calidad de los modelos predictivos anteriormente presentados.

```
x1<-pROC::roc(titanic_train2$Survived,predict(object =glm1,newdata = titanic_train2,type="res
ponse"))
x2<-pROC::roc(titanic_train2$Survived,predict(object =glm2,newdata = titanic_train2,type="res
ponse"))
x3<-pROC::roc(titanic_train2$Survived,predict(object =glm3,newdata = titanic_train2,type="res
ponse"))
x4<-pROC::roc(titanic_train2$Survived,predict(object =glm4,newdata = titanic_train2,type="res
ponse"))
x5<-pROC::roc(titanic_train2$Survived,predict(object =glm5,newdata = titanic_train2,type="res
ponse"))
x6<-pROC::roc(titanic_train2$Survived,predict(object =glm6,newdata = titanic_train2,type="res
ponse"))
x7<-pROC::roc(titanic_train2$Survived,predict(object =glm7,newdata = titanic_train2,type="res
ponse"))

pROC::plot.roc(x1,colorize=TRUE,col='red')
pROC::plot.roc(x2, add=TRUE,colorize=TRUE,col='blue')
pROC::plot.roc(x3, add=TRUE,colorize=TRUE,col='green')
pROC::plot.roc(x4, add=TRUE,colorize=TRUE,col='orange')
pROC::plot.roc(x5, add=TRUE,colorize=TRUE,col='black')
pROC::plot.roc(x6, add=TRUE,colorize=TRUE,col='purple')
pROC::plot.roc(x7, add=TRUE,colorize=TRUE,col='yellow')
```



Antes de comenzar a analizar el resultado obtenido por cada uno de los modelos definidos, hay que comprender que se representa en él. El eje y del gráfico representa la sensibilidad o verdaderos positivos, mientras que en el eje x se representa la especificidad o los falsos positivos. Partiendo de estos conceptos se puede afirmar que, cuanto más se aproximen los valores de un modelo a la esquina superior izquierda del gráfico (coordenada (0,1)) mejor será, puesto que esto indicará que el modelo no arroja ningún falso negativo y ningún falso positivo.

Tal y como se puede observar el grafico muestra varias curvas ROC con una área similar debajo de ella. Indicar que a mayor área debajo de la curva mejor es el modelo generado puesto que esto refleja que los valores calculados por el modelo se aproximan al 100% de sensibilidad (ningún falso negativo) y un 100% de especificidad (ningún falso positivo). Dicho de otro modo cuanto mayor sea el área debajo de la curva ROC, mejor será el modelo.

En base a lo anteriormente descrito si ahora se comprueba la curva ROC de los modelos graficados con menor área, se puede ver como éste corresponde al modelo 1 (curva roja). Esto coincide con los resultados calculados mediante la medida AIC, donde dicho modelo era el que tenía un valor mayor al resto mientras que los otros eran muy similares, aspecto que se ve también reflejado en el gráfico viendo que las curvas de los demás modelos prácticamente están superpuestas entre ellas.

Para verificar y comprobar que el área mayor corresponde a la del modelo 4 y la menor a la del modelo 1 se utiliza la función auc() que devuelve el valor del área debajo de la curva ROC.

```
#MODELO 1
pROC::auc(x1)
```

```
## Area under the curve: 0.6814
```

```
#MODELO 2
pROC::auc(x2)
```

```
## Area under the curve: 0.8328
```

```
#MODELO 3
pROC::auc(x3)
```

```
## Area under the curve: 0.847
```

```
#MODELO 4
pROC::auc(x4)
```

```
## Area under the curve: 0.8541
```

```
#MODELO 5
pROC::auc(x5)
```

```
## Area under the curve: 0.8543
```

```
#MODELO 6
pROC::auc(x6)
```

```
## Area under the curve: 0.8545
```

```
#MODELO 7
pROC::auc(x7)
```

```
## Area under the curve: 0.8569
```

Tal y como se puede apreciar en los resultados el modelo 1 es el que tiene un valor de área menor mientras que el resto de modelos prácticamente tienen el mismo valor. En este caso se obtiene que el modelo con mayor área es el 7.

5. Representación de los resultados a partir de tablas y gráficas.

Una vez calculado el modelo predictivo se aplica al bloque de datos **test**.

```
#Cálculo de los valores resultados con el modelo de regresión:
p<-predict(object =glm4,newdata = titanic_test2,type="response")

#Aplicación del umbral de discriminación del 75% a los resultados del modelo:
prediocciones <- ifelse(p > 0.75, 1, 0)

# Transformar en data.frame la lista de predicciones del modelo de regresión
lista<- data.frame(matrix(unlist(prediocciones), nrow=418, byrow=T),stringsAsFactors=FALSE)
names(lista)<-c("Survived")

#Carga de resultados en el conjunto de datos test2
titanic_test2$Survived<-lista$Survived

##Se generan los archivos test y entreno con los datos depurados
write.csv(titanic_test2,file="titanic_test_tratado.csv",row.names=FALSE)
write.csv(titanic_train2,file="titanic_train_tratado.csv",row.names=FALSE)
```

Con los datos finalmente cargados en el conjunto de datos de test, finalmente se vuelven a juntar los dos grupos de datos para su análisis final.

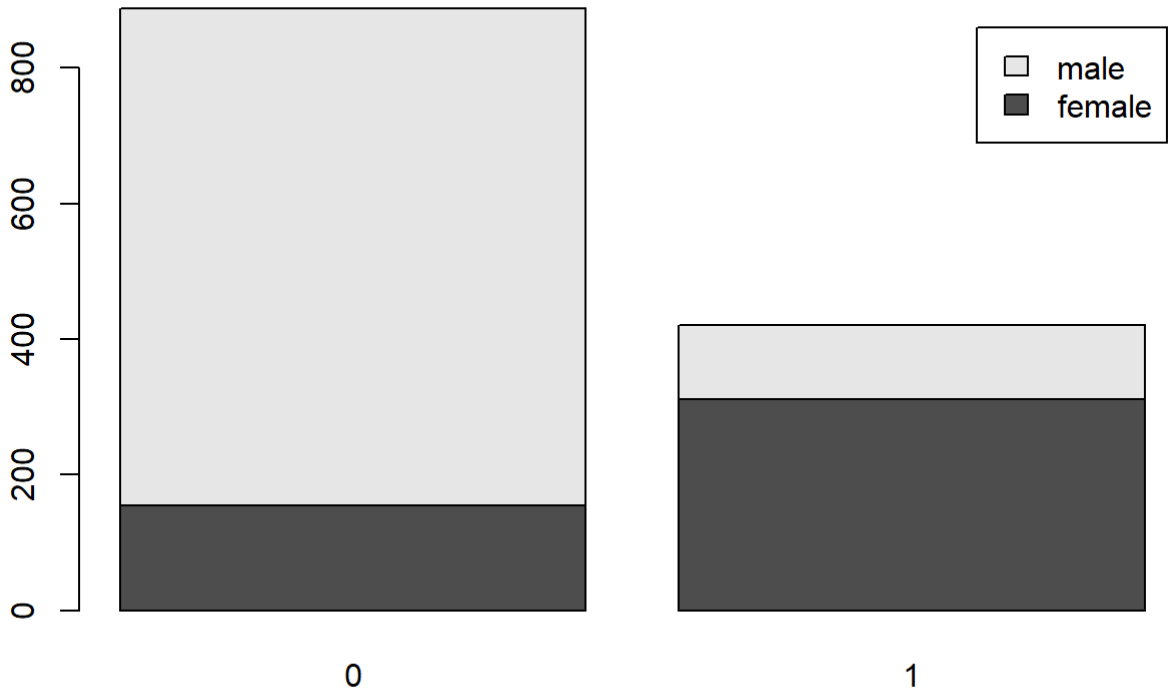
```
#Tabla general
titanic2<-rbind(titanic_train2,titanic_test2)
```

Análisis de los resultados:

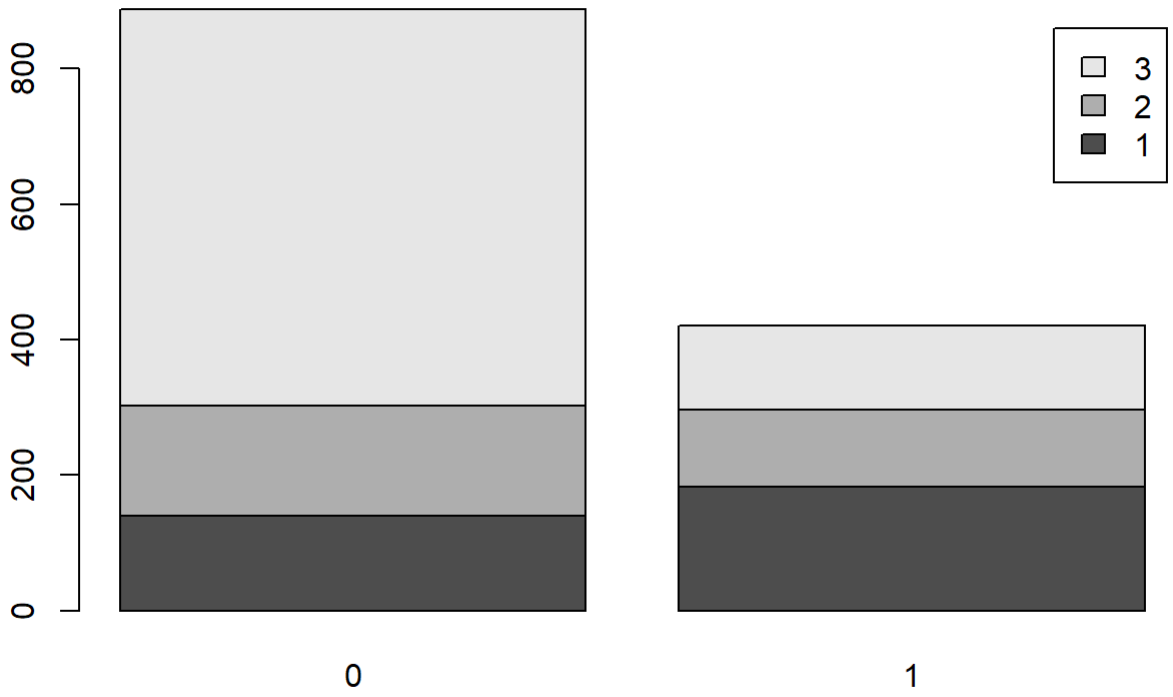
```
#Tabla supervivientes
table(titanic2$Survived)
```

```
##
##   0   1
## 888 421
```

```
#Supervivientes según sexo
barplot(table(titanic2$Sex,titanic2$Survived),legend=TRUE)
```




```
#Supervivientes según clase
barplot(table(titanic2$Pclass,titanic2$Survived),legend=TRUE)
```



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

En base a los gráficos obtenidos en el apartado anterior se puede apreciar la influencia de los factores que se notaron en el análisis del modelo predictivo:

- Dentro de los registros de supervivientes el número de tripulantes masculinos es inferior al femenino. De forma contrario en el caso de los registros fallecidos, el tripulante de sexo masculino es mayor al femenino.
- A medida que la clase del pasajero mejora, el número de registros supervivientes también incrementa (supervivientes primera clase > supervivientes tercera clase). En el caso de los registros fallecidos, estos se dan en mayor proporción en usuarios de tercera y segunda clase.

Como conclusiones finales se puede decir que se ha llegado a un modelo predictivo que ha permitido predecir a partir de unos parámetros que definen el usuario del registro, si el pasajero sobreviviría o no al accidente respondiendo así a la pregunta formulada al inicio del ejercicio.

Indicar que la calidad del modelo es mejorable y es a partir de este punto en el que toca continuar iterando y realizando modificaciones en el modelo definido con la finalidad de mejorar la calidad de predicción ajustándola cada vez más.