Tipologia y ciclo de vida de los datos: Práctica 2

Autor: Sandra Garrido Romero

Diciembre 2019

Table of Contents

Presentación

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas. Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github (https://github.com) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo. Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github. Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía: * Ejemplo: https://github.com/Bengis/nba-gap-cleaning * Ejemplo complejo (archivo adjunto).

Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Resolución

Descripción del dataset

El conjunto de datos a utilizar se ha descargado desde este enlace https://www.kaggle.com/c/titanic/data de la página web de *Kaggle*. Este dataset se conforma de 12 variables (columnas) y 1309 pasajeros (filas). El conjunto de datos se ha conseguido uniendo los conjuntos de entrenamiento y test. Los campos del conjunto de datos son los siguientes:

- **PassengerId**: Identificador de cada uno de los pasajeros.
- **Survived**: Indica si el pasajero sobrevivió (1) o no (0).
- **Pclass**: Clase que figuraba en el ticket: primera clase (1), segunda clase (2) o tercera clase (3).

- **Name** : Nombre del pasajero.
- **Sex** : Sexo del pasajero (male, female).
- **Age** : Edad del pasajero.
- **SibSp** : Relaciones familiares del siguiente tipo: hermanos, hermanastros, esposos o esposas.
- **Parch** : Relaciones familiares del siguiente tipo: madre, padre, hijos o hijastros.
- **Ticket** : Número de ticket.
- **Fare** : Precio del ticket.
- **Cabin** : Número de cabina.
- **Embarked** : Puerto de embarque: Cherbourg (C), Queenstown (Q) o Southampton (S).

Importancia y objetivos de los análisis

Este conjunto de datos puede ayudar a resolver qué variables son las que influyen en si un pasajero sobrevivió al hundimiento del Titanic o no.

Integración y selección de los datos

En primer lugar, se va a proceder a leer el fichero de datos obtenido de internet. Se cuenta con dos ficheros CSV, por un lado, el fichero de datos de entrenamiento y por otro, el fichero de datos test. Se van a importar los dos archivos y se van a unir en un sólo *data.frame*:

```
#Lectura del fichero test
test <- read.csv('test.csv',stringsAsFactors = FALSE)
#Letura del fichero entrenamiento
train <- read.csv('train.csv',stringsAsFactors = FALSE)
# Unión de los dos conjuntos importados en uno solo
library(dplyr)
datos <- bind_rows(train,test)
```

Se observa una parte de los datos y el tipo de estos:

```
head(datos)
```

```
##   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
## 1                             Braund, Mr. Owen Harris   male  22     1
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
## 3                              Heikkinen, Miss. Laina female  26     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
```

```
## 5                            Allen, Mr. William Henry   male  35     0
## 6                                    Moran, Mr. James   male  NA     0
##   Parch           Ticket    Fare Cabin Embarked
## 1     0        A/5 21171  7.2500              S
## 2     0         PC 17599 71.2833   C85        C
## 3     0 STON/O2. 3101282  7.9250              S
## 4     0           113803 53.1000  C123        S
## 5     0           373450  8.0500              S
## 6     0           330877  8.4583              Q
```

```
str(datos)
```

```
## 'data.frame':    1309 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)" "Heikkinen, Miss. Laina" "Futrelle,
Mrs. Jacques Heath (Lily May Peel)" ...
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282"
"113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Más adelante se tratará el tipo de los datos.

Limpieza de los datos

En este apartado se va a proceder a limpiar los datos de manera que no contengan valores vacíos o valores extremos.

Ceros y elementos vacíos

En primer lugar, se procede a comprobar si existen valores nulos en el conjunto:

```
colSums(is.na(datos))
```

```
## PassengerId    Survived      Pclass        Name         Sex
Age
##           0         418           0           0           0
263
##       SibSp       Parch      Ticket        Fare       Cabin
Embarked
##           0           0           0           1           0
0
```

Existen valores nulos en las variables *Survived*, *Age* y *Fare*.

Se buscan valores vacíos:

```
colSums(datos=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex
Age
##           0          NA           0           0           0
NA
##       SibSp       Parch      Ticket        Fare       Cabin
Embarked
##           0           0           0          NA        1014
2
```

Las siguientes columnas presentan valores vacíos: *Cabin* y *Embarked*.

Los valores nulos de la variable *Edad* se van a rellenar con la media del resto de edades.

```
datos$Age[is.na(datos$Age)] <- mean(datos$Age,na.rm=T)
```

Los valores de la variable *Survived* nulos corresponden a los registros procedentes del archivo de datos test.

Los valores de *Cabin* y *Fare* vacíos no se van a tratar ya que es una variable que no tiene repercusión en el análisis.

Los valores *Embarked* vacíos se van a sustituir por "S" que representa el puerto de salida del barco.

```
datos$Embarked[datos$Embarked==""]="S"
```

Algunos de los tipos que R ha asignado a los datos se deben cambiar de tipo *int* a tipo *factor* ya que son variables que clasifican a los datos.

```
datos$Survived<-as.factor(datos$Survived)
datos$Pclass<-as.factor(datos$Pclass)
datos$Sex<-as.factor(datos$Sex)
datos$Embarked<-as.factor(datos$Embarked)
```

Se comprueba que la conversión se ha realizado correctamente:

```
str(datos)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2
...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John
Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle,
Mrs. Jacques Heath (Lily May Peel)" ...
```

```
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1
1 ...
##  $ Age        : num  22 38 26 35 35 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282"
"113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1
...
```

Valores extremos

Los valores extremos son aquellos que distan mucho del resto de valores del conjunto de datos. Se va a utilizar la función *boxplots.stats()* de R para determinar qué valores extremos contiene cada variable. Los valores extremos se van a buscar en las variables numéricas.

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

```
##   [1]  2.00 58.00 55.00  2.00 66.00 65.00  0.83 59.00 71.00 70.50
2.00
##  [12] 55.50  1.00 61.00  1.00 56.00  1.00 58.00  2.00 59.00 62.00
58.00
##  [23] 63.00 65.00  2.00  0.92 61.00  2.00 60.00  1.00  1.00 64.00
65.00
##  [34] 56.00  0.75  2.00 63.00 58.00 55.00 71.00  2.00 64.00 62.00
62.00
##  [45] 60.00 61.00 57.00 80.00  2.00  0.75 56.00 58.00 70.00 60.00
60.00
##  [56] 70.00  0.67 57.00  1.00  0.42  2.00  1.00 62.00  0.83 74.00
56.00
##  [67] 62.00 63.00 55.00 60.00 60.00 55.00 67.00  2.00 76.00 63.00
1.00
##  [78] 61.00 60.50 64.00 61.00  0.33 60.00 57.00 64.00 55.00  0.92
1.00
##  [89]  0.75  2.00  1.00 64.00  0.83 55.00 55.00 57.00 58.00  0.17
59.00
## [100] 55.00 57.00
```

Los valores extremos que aparecen en la variable *Age* se van a dejar como están ya que aparecen debido a que las edades de los pasajeros son muy variadas dado la existencia de bebés y personas mayores a bordo del barco.

```
boxplot.stats(datos$SibSp)$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
## [36] 5 4 3 4 8 4 3 4 8 4 8 3 4 5 3 4 8 4 8 4 3 3
```

Los valores extremos de *SibSp* tampoco se van a tratar ya que es perfectamente válido que haya personas con muchos familiares a bordo.

```
boxplot.stats(datos$Parch)$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
##  [36] 2 1 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
##  [71] 1 2 1 2 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
## [106] 2 3 4 1 2 1 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
## [141] 2 1 1 2 5 2 1 1 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
## [176] 1 2 1 2 3 1 2 1 2 2 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
## [211] 1 5 2 1 1 1 1 3 1 2 2 1 2 1 2 1 2 4 1 1 2 1 1 1 4 6 2 3 1 1 2 2
2 1 1
## [246] 2 5 2 3 2 1 1 1 2 1 2 2 2 1 2 1 1 2 1 2 1 2 1 2 2 1 1 1 1 1 2 1
1 2 1
## [281] 1 1 2 1 2 9 1 1 1 2 2 2 1 9 1 1 2 2 1 1 2 1 1 1 1 1 1
```

Con la variable *Parch* ocurre lo mismo que con la anterior.

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

```
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750
##   [8]  73.5000 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000
##  [15]  66.6000  69.5500  69.5500 146.5208  69.5500 113.2750  76.2917
##  [22]  90.0000  83.4750  90.0000  79.2000  86.5000 512.3292  79.6500
##  [29] 153.4625 135.6333  77.9583  78.8500  91.0792 151.5500 247.5208
##  [36] 151.5500 110.8833 108.9000  83.1583 262.3750 164.8667 134.5000
##  [43]  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000 263.0000
##  [50]  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
##  [64]  91.0792  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000
##  [71] 221.7792 106.4250  71.0000 106.4250 110.8833 227.5250  79.6500
##  [78] 110.8833  79.6500  79.2000  78.2667 153.4625  77.9583  69.3000
##  [85]  76.7292  73.5000 113.2750 133.6500  73.5000 512.3292  76.7292
##  [92] 211.3375 110.8833 227.5250 151.5500 227.5250 211.3375 512.3292
##  [99]  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583 211.3375
## [106]  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500
## [113]  89.1042 164.8667  69.5500  83.1583  82.2667 262.3750  76.2917
## [120] 263.0000 262.3750 262.3750 263.0000 211.5000 211.5000 221.7792
## [127]  78.8500 221.7792  75.2417 151.5500 262.3750  83.1583 221.7792
## [134]  83.1583  83.1583 247.5208  69.5500 134.5000 227.5250  73.5000
## [141] 164.8667 211.5000  71.2833  75.2500 106.4250 134.5000 136.7792
## [148]  75.2417 136.7792  82.2667  81.8583 151.5500  93.5000 135.6333
## [155] 146.5208 211.3375  79.2000  69.5500 512.3292  73.5000  69.5500
```

```
## [162]  69.5500 134.5000  81.8583 262.3750  93.5000  79.2000 164.8667
## [169] 211.5000  90.0000 108.9000
```

Por último, los valores extremos de la variable *Fare* también se van a dejar como están porque el precio entre los billetes de primera y tercera clase varían mucho.

Se exporta el conjunto de datos preprocesado a un nuevo fichero: Se exporta el conjunto de datos a un nuevo archivo.

```
write.csv(datos,"datos_clean.csv")
```

Análisis de los datos

Selección de los grupos de datos a analizar

En primer lugar, se tienen los grupos de datos divididos en datos de entrenamiento y datos test tal cual se han descargado de la página web.

Para obtener más agrupaciones de los datos, se van a dividir según el valor de distintas variables.

```
# Agrupación por supervivientes
datos.survived<-datos[datos$Survived=="1",]
datos.nosurvived<-datos[datos$Survived=="1",]

# Agrupación por clase
datos.primera<-datos[datos$Pclass=="1",]
datos.segunda<-datos[datos$Pclass=="2",]
datos.tercera<-datos[datos$Pclass=="3",]

# Agrupación por sexo
datos.mujer<-datos[datos$Sex=="female",]
datos.hombre<-datos[datos$Sex=="male",]
```

Comprobación de la normalidad y homogeneidad de la varianza

La comprobación de la normalidad de las variables numéricas se llevará a cabo mediante la prueba de _Pearson__Wilk_. En este caso, las variables que se estudiarán son *Age*, *SibSp*, *Parch* y *Fare*.

Se parte de la hipótesis nula de que las muestras provienen de una distribución normal, y de la hipótesis alternativa de que las muestras no provienen de distribuciones normales.

El nivel de significancia que se utilizará es 0,05. Si el p-valor obtenido en la prueba es menor que 0,05 entonces se rechara la hipótesis nula.

```
shapiro.test(datos[,6])

##
##  Shapiro-Wilk normality test
```

```
##
## data:  datos[, 6]
## W = 0.95758, p-value < 2.2e-16
```

```
shapiro.test(datos[,7])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos[, 7]
## W = 0.51108, p-value < 2.2e-16
```

```
shapiro.test(datos[,8])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos[, 8]
## W = 0.49797, p-value < 2.2e-16
```

```
shapiro.test(datos[,10])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos[, 10]
## W = 0.52782, p-value < 2.2e-16
```

Además, se va a aplicar la prueba de *Pearson* para comprobar si se obtienen los mismo resultados.

```
library("nortest")
pearson.test(datos[,6])
```

```
##
##  Pearson chi-square normality test
##
## data:  datos[, 6]
## P = 2232.7, p-value < 2.2e-16
```

```
pearson.test(datos[,7])
```

```
##
##  Pearson chi-square normality test
##
## data:  datos[, 7]
## P = 23461, p-value < 2.2e-16
```

```
pearson.test(datos[,8])
```

```
##
##  Pearson chi-square normality test
##
```

```
## data:  datos[, 8]
## P = 27466, p-value < 2.2e-16
```

```
pearson.test(datos[,10])
```

```
##
##  Pearson chi-square normality test
##
## data:  datos[, 10]
## P = 6901.1, p-value < 2.2e-16
```

Todas las pruebas realizadas han obtenido un p-valor menor al nivel de significancia prefijado, por lo que se rechaza la hipótesis nula en todos los casos. Ninguna de las variables estudiadas sigue una distribución normal.

Aún así, al contar con más de 30 registros, por el teorema central del límite, los datos se pueden normalizar.

Se va a estudiar la homogeneidad de varianzas en la variable *Age* en función de los grupos que se han creado anteriormente. Se parte de la hipótesis nula de que las varianzas de las muestras son iguales, y de la hipótesis alternativa de que no son iguales. El nivel de significancia utilizado será 0,05.

```
# En función del sexo
fligner.test(datos$Age ~ datos$Sex, data = datos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  datos$Age by datos$Sex
## Fligner-Killeen:med chi-squared = 4.3943, df = 1, p-value =
## 0.03606
```

```
# En función de si sobrevivió o no
fligner.test(datos$Age ~ datos$Survived, data = datos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  datos$Age by datos$Survived
## Fligner-Killeen:med chi-squared = 5.5164, df = 1, p-value =
## 0.01884
```

```
# En función de la clase en la que viajaban
fligner.test(datos$Age ~ datos$Pclass, data = datos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  datos$Age by datos$Pclass
## Fligner-Killeen:med chi-squared = 69.158, df = 2, p-value =
## 9.604e-16
```

Los p-valor obtenidos en cada una de las pruebas son menores a 0,05 por lo que se rechaza la hipótesis de que las varianzas son homogéneas.

Pruebas estadisticas

Se obtiene una matriz de porcentajes de frecuencias de la variable *Sex* en función de si el pasajero sobrevivió o no.

```
filas=dim(datos)[1]
t<-table(datos[1:filas,]$Sex,datos[1:filas,]$Survived)
for (i in 1:dim(t)[1]){
    t[i,]<-t[i,]/sum(t[i,])*100
}
t
```

```
##
##                 0        1
##   female 25.79618 74.20382
##   male   81.10919 18.89081
```

En la tabla se puede ver que las mujeres tenían más probabilidad de sobrevivir que los hombres.

Como se dispone de un conjunto de datos de entrenamiento y otro de test, se va a crear un árbol de decisión para comprobar si se predicen los datos correctamente.

```
y <- datos[,2]
X <- select(datos,Pclass,Sex,Embarked)
trainX <- X[1:891,]
trainy <- y[1:891]
model <- C50::C5.0(trainX, trainy,rules=TRUE )
summary(model)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Fri Dec 27 19:38:43 2019
## -----------------------------
##
## Class specified by attribute `outcome'
##
## Read 891 cases (4 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (577/109, lift 1.3)
##  Sex = male
##  ->  class 0  [0.810]
##
```

```
## Rule 2: (491/119, lift 1.2)
##  Pclass = 3
##  ->  class 0  [0.757]
##
## Rule 3: (170/9, lift 2.5)
##  Pclass in {1, 2}
##  Sex = female
##  ->  class 1  [0.942]
##
## Rule 4: (109/18, lift 2.2)
##  Sex = female
##  Embarked in {C, Q}
##  ->  class 1  [0.829]
##
## Default class: 0
##
##
## Evaluation on training data (891 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       4  168(18.9%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     523    26    (a): class 0
##     142   200    (b): class 1
##
##
##  Attribute usage:
##
##   90.12% Sex
##   74.19% Pclass
##   12.23% Embarked
##
##
## Time: 0.0 secs
```

Se utilizan los datos pertenecientes al conjunto test y se comparan con los que se encuentran en el archivo *gender_submission.csv*. La matriz de confusión servirá para comprobar si el modelo creado clasifica correctamente los datos.

```
testX <- X[892:1309,]
testy <- read.csv('gender_submission.csv',stringsAsFactors = FALSE)
testy<-testy$Survived
predicted_model <- predict( model, testX, type="class" )
```

```
mat_conf<-table(testy,Predicted=predicted_model)
mat_conf
```

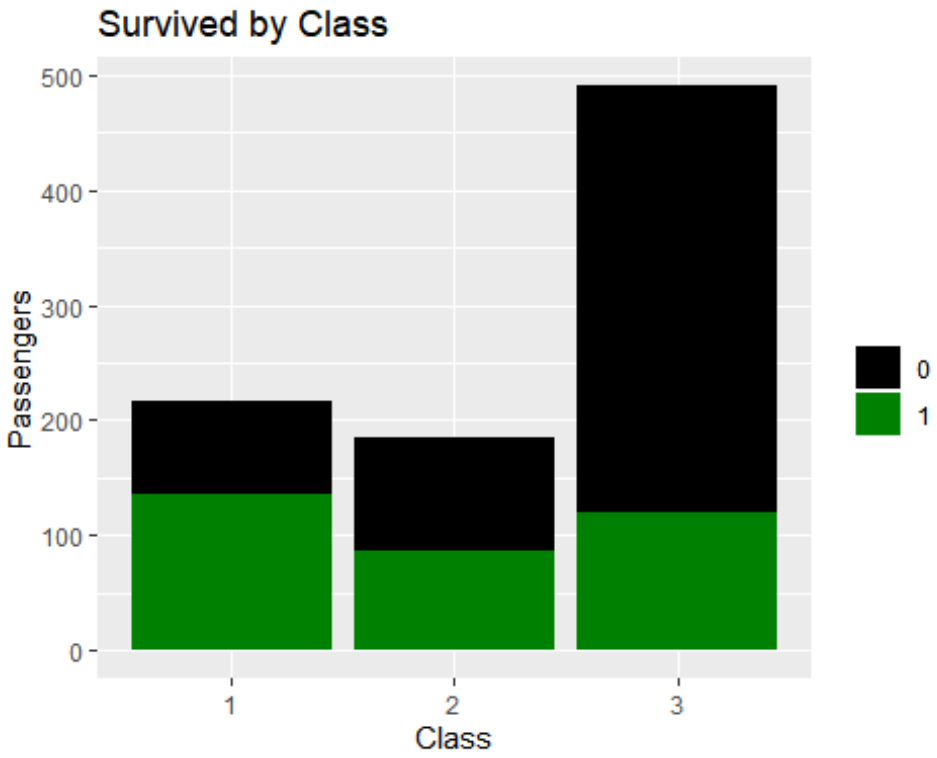
```
##      Predicted
## testy   0   1
##     0 266   0
##     1  41 111
```

Los no supervivientes se predicen correctamente pero, hay fallos en los supervivientes.

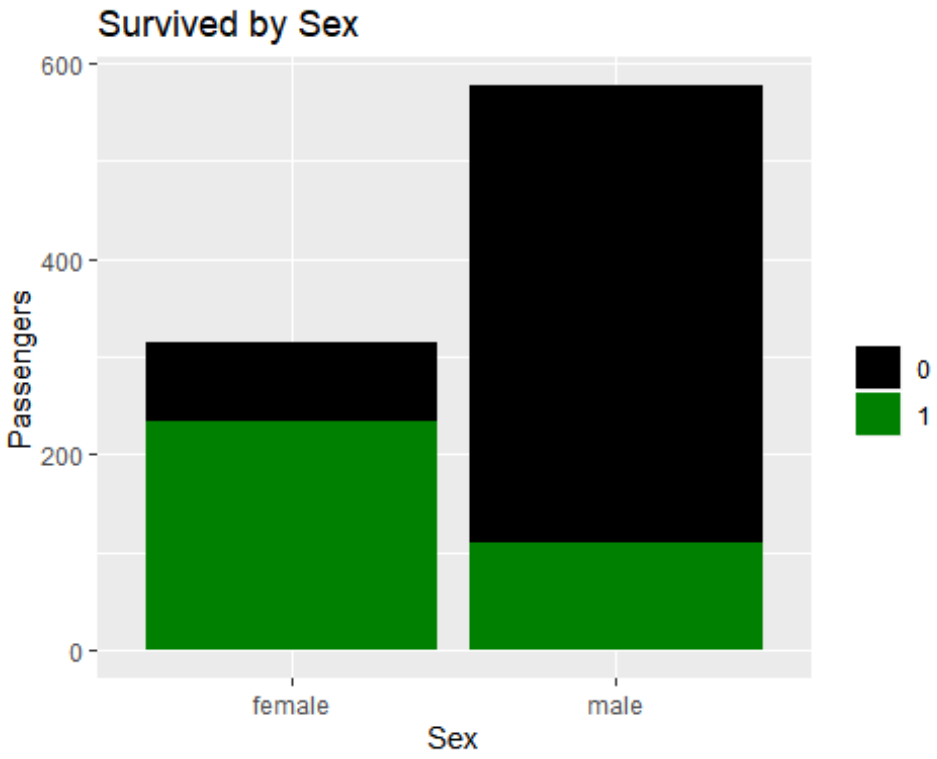
Representación de resultados

Se van a representar las variables de tipo *factor* en función de si el personaje sobrevivió o no, para ver si concuerda con los resultados obtenidos.

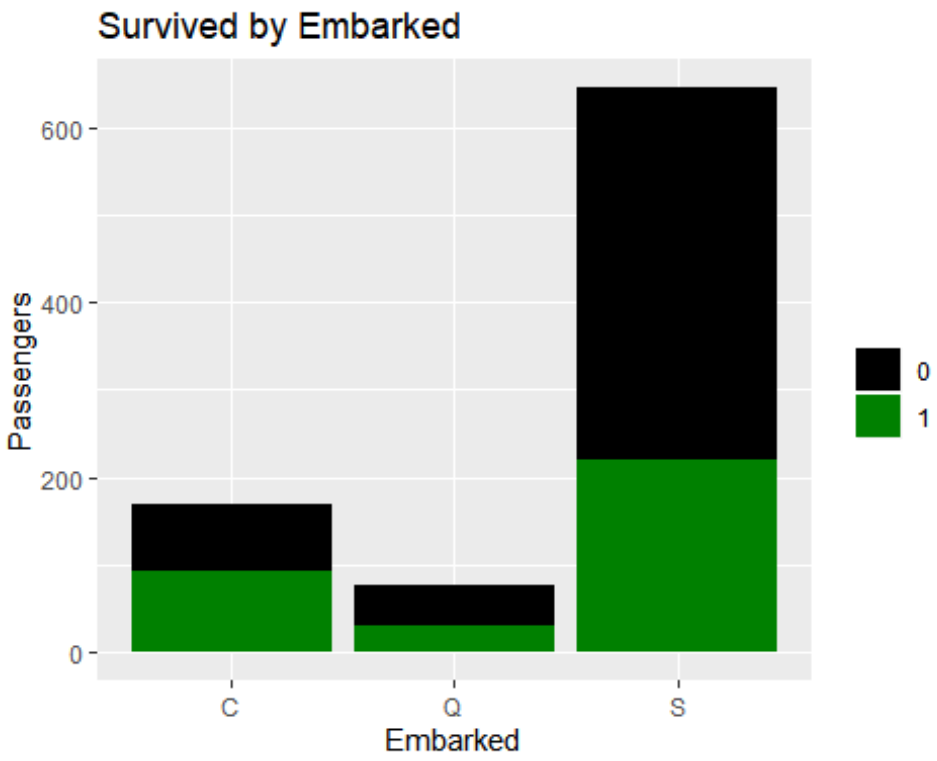
```
library(ggplot2)
train<-datos[1:891,]
ggplot(train,aes(Pclass,fill=Survived))+geom_bar() +labs(x="Class",
y="Passengers")+ guides(fill=guide_legend(title=""))+
scale_fill_manual(values=c("black","#008000"))+ggtitle("Survived by
Class")
```




```
ggplot(train,aes(Sex,fill=Survived))+geom_bar() +labs(x="Sex",
y="Passengers")+ guides(fill=guide_legend(title=""))+
scale_fill_manual(values=c("black","#008000"))+ggtitle("Survived by Sex")
```




```
ggplot(train,aes(Embarked,fill=Survived))+geom_bar() +labs(x="Embarked",
y="Passengers")+ guides(fill=guide_legend(title=""))+
scale_fill_manual(values=c("black","#008000"))+ggtitle("Survived by
Embarked")
```



Conclusiones