Tipología y ciclo de Vida de los datos. PRA2

04/01/2020

Table of Contents

Importación previa de librerías

library(ggplot2)  
library(dplyr)  
library(kableExtra)  
library(VIM)  
library(arules)  
library(car)

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

El dataset corresponde a la colección de datos de entrenamiento parte de una competición activa de Kaggle y el objeto de esta competición es la realización de análisis predictivo sobre qué pasajeros sobrevivieron al naufragio del Titanic.

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

## Carga de datos y análisis inicial

Para empezar cargamos los datos. No hará falta integrarlos porque tenemos un único origen de datos, por lo que nos centraremos en el análisis y limpieza de estos.

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

Hacemos una primera revisión de los datos, mirando la dimensión del data frame importado y las clases de cada variable del mismo

# Breve análisis de los datos   
# Dimensiones de la base de datos mediante la función dim(). Obtenemos que disponemos de 891 registros o pasajeros (filas) y 12 variables (columnas).   
dim(data.train)

## [1] 891 12

# Examinamos el tipo de datos con los que R ha interpretado cada variable.  
sapply(data.train,class)

## PassengerId Survived Pclass Name Sex Age   
## "integer" "integer" "integer" "character" "character" "numeric"   
## SibSp Parch Ticket Fare Cabin Embarked   
## "integer" "integer" "character" "numeric" "character" "character"

Vemos que nuestro dataset no es muy extenso, con tan sólo 891 individuos y 12 variables diferentes para trabajar. Como referencia, ponemos un breve diccionario que explica cada variable

## Data Dictionary

PassengerId -> id of de passenger

survived -> 0 = No; 1 = Yes

pclass -> Passenger Class 1 = 1st; 2 = 2nd; 3 = 3rd

name -> First and Last Name

sex -> Sex

age -> Age

sibsp -> Number of Siblings/Spouses Aboard

parch -> Number of Parents/Children Aboard

ticket -> Ticket Number

fare -> Passenger Fare

cabin -> Cabin

embarked -> Port of Embarkation C = Cherbourg; Q = Queenstown; S = Southampton

## Formato de variables

Examinamos distribución de valores por variables para ver si hay alguna que esté en un formato inadecuado

summary(data.train)

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

Observamos que tenemos 177 NA’s en la variable Age, pero estos valores perdidos los trataremos en otro apartado. Reformateamos las siguientes variables para trabajar mejor con ellas

|  |  |  |
| --- | --- | --- |
| Variable | Formato origen | Formato destino |
| Survived | entero | factor |
| Pclass | entero | factor |
| Sex | string | factor |
| Ticket | string | factor |
| Cabin | string | factor |
| Embarked | string | factor |

#Survived de entero a factor  
data.train$Survived <- factor(data.train$Survived, levels=c(0,1), labels=c("No", "Sí"))  
levels(data.train$Survived)

## [1] "No" "Sí"

#Pclass de entero a factor  
data.train$Pclass <- factor(data.train$Pclass, levels=c(1,2,3), labels=c("Primera clase", "Segunda clase", "Tercera clase"))  
levels(data.train$Pclass )

## [1] "Primera clase" "Segunda clase" "Tercera clase"

#R ha interpretado la variable Sex como un string, la cambiamos a factor  
data.train$Sex<- factor(data.train$Sex)  
levels(data.train$Sex)

## [1] "female" "male"

#R ha interpretado la variable Ticket como un string, la cambiamos a factor  
data.train$Ticket<- factor(data.train$Ticket)  
head(levels(data.train$Ticket))

## [1] "110152" "110413" "110465" "110564" "110813" "111240"

# R ha interpretado la variable Cabin como un string, la cambiamos a factor  
data.train$Cabin<- factor(data.train$Cabin)  
head(levels(data.train$Cabin))

## [1] "" "A10" "A14" "A16" "A19" "A20"

# R ha interpretado la variable Embarked como un string, la cambiamos a factor  
data.train$Embarked<- factor(data.train$Embarked, levels=c("C", "Q", "S"),labels=c("Cherbourg", "Queenstown", "Southampton"))  
levels(data.train$Embarked)

## [1] "Cherbourg" "Queenstown" "Southampton"

Revisamos cómo ha quedado todo después de los cambios de formato

head(data.train)

## PassengerId Survived Pclass  
## 1 1 No Tercera clase  
## 2 2 Sí Primera clase  
## 3 3 Sí Tercera clase  
## 4 4 Sí Primera clase  
## 5 5 No Tercera clase  
## 6 6 No Tercera clase  
## Name Sex Age SibSp Parch  
## 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 Southampton  
## 2 PC 17599 71.2833 C85 Cherbourg  
## 3 STON/O2. 3101282 7.9250 Southampton  
## 4 113803 53.1000 C123 Southampton  
## 5 373450 8.0500 Southampton  
## 6 330877 8.4583 Queenstown

sapply(data.train,class)

## PassengerId Survived Pclass Name Sex Age   
## "integer" "factor" "factor" "character" "factor" "numeric"   
## SibSp Parch Ticket Fare Cabin Embarked   
## "integer" "integer" "factor" "numeric" "factor" "factor"

# 3. Limpieza de los datos

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

Miramos el número de valores desconocidos y valores vacios por campo

sapply(data.train, function(x) sum(is.na(x)))

## PassengerId Survived Pclass Name Sex Age   
## 0 0 0 0 0 177   
## SibSp Parch Ticket Fare Cabin Embarked   
## 0 0 0 0 0 2

sapply(data.train, function(x) sum(ifelse(x=="", 1,0)))

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

En resumen vemos que tenemos 177 NAs en Age, 687 campos vacíos en Cabin y 2 en Embarked.

Cambiamos los campos vacíos por NAs, pues no tenemos ningún motivo para diferenciar estos de los NAs y tratarlos diferente.

data.train$Cabin[data.train$Cabin==""]<- NA  
data.train$Cabin[data.train$Embarked==""]<- NA

Imputaremos los valores que faltan basándonos en la similitud o diferencia entre los registros: la imputación basada en k vecinos más próximos

Imputamos los valores para Age y Embarked, en la variable Cabin hay demasiada poca información como para haCer imputaciones

suppressWarnings(suppressMessages(library(VIM)))  
data.train$Age <- kNN(data.train[, 2:12] )$Age  
data.train$Embarked <- kNN(data.train[, 2:12])$Embarked  
summary(data.train)

## PassengerId Survived Pclass Name Sex   
## Min. : 1.0 No:549 Primera clase:216 Length:891 female:314   
## 1st Qu.:223.5 Sí:342 Segunda clase:184 Class :character male :577   
## Median :446.0 Tercera clase:491 Mode :character   
## Mean :446.0   
## 3rd Qu.:668.5   
## Max. :891.0   
##   
## Age SibSp Parch Ticket   
## Min. : 0.42 Min. :0.000 Min. :0.0000 1601 : 7   
## 1st Qu.:20.00 1st Qu.:0.000 1st Qu.:0.0000 347082 : 7   
## Median :28.00 Median :0.000 Median :0.0000 CA. 2343: 7   
## Mean :29.44 Mean :0.523 Mean :0.3816 3101295 : 6   
## 3rd Qu.:38.00 3rd Qu.:1.000 3rd Qu.:0.0000 347088 : 6   
## Max. :80.00 Max. :8.000 Max. :6.0000 CA 2144 : 6   
## (Other) :852   
## Fare Cabin Embarked   
## Min. : 0.00 B96 B98 : 4 Cherbourg :170   
## 1st Qu.: 7.91 C23 C25 C27: 4 Queenstown : 77   
## Median : 14.45 G6 : 4 Southampton:644   
## Mean : 32.20 C22 C26 : 3   
## 3rd Qu.: 31.00 D : 3   
## Max. :512.33 (Other) :186   
## NA's :687

Una vez resuelta la problemática de los valores vacios vemos cómo se distribuyen los valores de la edad y discretizamos esta variable para facilitar su análisis

summary(data.train$Age)

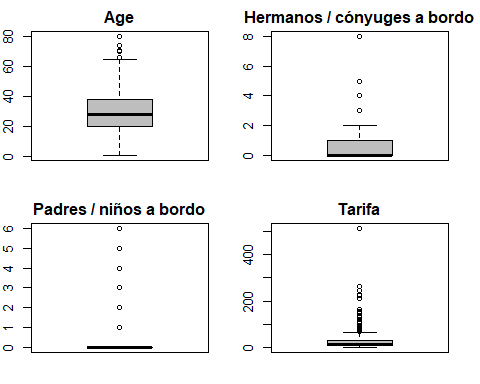
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.42 20.00 28.00 29.44 38.00 80.00

# Discretizamos  
data.train$AgeSegments <- cut(data.train$Age, breaks = c(0,10,20,30,40,50,60,70,110), labels = c("0-9", "10-19", "20-29", "30-39","40-49","50-59","60-69","70+"))

## 3.2. Identificación y tratamiento de valores extremos.

Representamos un diagrama de caja por cada variable para ver qué valores distan mucho del rango intercuartílico (la caja) en las variables numéricas

par(mar = c(2, 2, 2, 2))  
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE),widths=c(1,1,1), heights=c(1,1,1))  
boxplot(data.train$Age,main="Age", col="gray")  
boxplot(data.train$SibSp,main="Hermanos / cónyuges a bordo", col="gray")  
boxplot(data.train$Parch,main="Padres / niños a bordo", col="gray")  
boxplot(data.train$Fare,main="Tarifa", col="gray")



En ninguno de los casos los valores extremos que quedan fuera de los rangos parecen valores que no sean razonables. Quizás el que pueda levantar más sospechas es el valor altísimo que detectamos en la tarifa

Utilizamos la función boxplots.stats() de R para identificar los Valores extremos de Age y sus posiciones. Al ser pocos visualizamos el resto de variables de las personas en estos valores extremos

values <- boxplot.stats(data.train$Age)$out  
idx <- which( data.train$Age %in% values)  
Age.outliers <- data.train[idx,]  
Age.outliers %>% kable(caption="Outliers en Age") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

Outliers en Age

PassengerId

Survived

Pclass

Name

Sex

Age

SibSp

Parch

Ticket

Fare

Cabin

Embarked

AgeSegments

34

34

No

Segunda clase

Wheadon, Mr. Edward H

male

66.0

0

0

C.A. 24579

10.5000

NA

Southampton

60-69

97

97

No

Primera clase

Goldschmidt, Mr. George B

male

71.0

0

0

PC 17754

34.6542

A5

Cherbourg

70+

117

117

No

Tercera clase

Connors, Mr. Patrick

male

70.5

0

0

370369

7.7500

NA

Queenstown

70+

494

494

No

Primera clase

Artagaveytia, Mr. Ramon

male

71.0

0

0

PC 17609

49.5042

NA

Cherbourg

70+

631

631

Sí

Primera clase

Barkworth, Mr. Algernon Henry Wilson

male

80.0

0

0

27042

30.0000

A23

Southampton

70+

673

673

No

Segunda clase

Mitchell, Mr. Henry Michael

male

70.0

0

0

C.A. 24580

10.5000

NA

Southampton

60-69

746

746

No

Primera clase

Crosby, Capt. Edward Gifford

male

70.0

1

1

WE/P 5735

71.0000

B22

Southampton

60-69

852

852

No

Tercera clase

Svensson, Mr. Johan

male

74.0

0

0

347060

7.7750

NA

Southampton

70+

Tras ver el resto de datos de estos pasajeros sigue pareciendo del todo razonable la edad registrada, por lo que decidimos no actuar sobre estos valores extremos

values <- boxplot.stats(data.train$Age)$out  
idx <- which( data.train$Age %in% values)  
Age.outliers <- data.train[idx,]  
Age.outliers %>% kable(caption="Outliers en Age") %>% kable\_styling(bootstrap\_options = c("striped", "hover"))

Outliers en Age

PassengerId

Survived

Pclass

Name

Sex

Age

SibSp

Parch

Ticket

Fare

Cabin

Embarked

AgeSegments

34

34

No

Segunda clase

Wheadon, Mr. Edward H

male

66.0

0

0

C.A. 24579

10.5000

NA

Southampton

60-69

97

97

No

Primera clase

Goldschmidt, Mr. George B

male

71.0

0

0

PC 17754

34.6542

A5

Cherbourg

70+

117

117

No

Tercera clase

Connors, Mr. Patrick

male

70.5

0

0

370369

7.7500

NA

Queenstown

70+

494

494

No

Primera clase

Artagaveytia, Mr. Ramon

male

71.0

0

0

PC 17609

49.5042

NA

Cherbourg

70+

631

631

Sí

Primera clase

Barkworth, Mr. Algernon Henry Wilson

male

80.0

0

0

27042

30.0000

A23

Southampton

70+

673

673

No

Segunda clase

Mitchell, Mr. Henry Michael

male

70.0

0

0

C.A. 24580

10.5000

NA

Southampton

60-69

746

746

No

Primera clase

Crosby, Capt. Edward Gifford

male

70.0

1

1

WE/P 5735

71.0000

B22

Southampton

60-69

852

852

No

Tercera clase

Svensson, Mr. Johan

male

74.0

0

0

347060

7.7750

NA

Southampton

70+

Analizamos ahora los extremos en el número de hermanos

unique(boxplot.stats(data.train$SibSp)$out)

## [1] 3 4 5 8

Los casos de 3 y 4 hermanos, aún siendo extremos los damos directamente por buenos, pues era muy frecuente ese número de hermanos en la época. Nos centramos en los casos de 5 y 8 hermanos para ver si son razonables

data.train[data.train$SibSp==5,c(3,4,6,9)]

## Pclass Name Age Ticket  
## 60 Tercera clase Goodwin, Master. William Frederick 11 CA 2144  
## 72 Tercera clase Goodwin, Miss. Lillian Amy 16 CA 2144  
## 387 Tercera clase Goodwin, Master. Sidney Leonard 1 CA 2144  
## 481 Tercera clase Goodwin, Master. Harold Victor 9 CA 2144  
## 684 Tercera clase Goodwin, Mr. Charles Edward 14 CA 2144

En el caso de 5 hermanos vemos que coinciden ticket y apellidos, por lo que los damos por buenos.

Analizamos el caso de 8 hermanos

data.train[data.train$SibSp==8,c(3,4,6,7,9)]

## Pclass Name Age SibSp Ticket  
## 160 Tercera clase Sage, Master. Thomas Henry 9 8 CA. 2343  
## 181 Tercera clase Sage, Miss. Constance Gladys 9 8 CA. 2343  
## 202 Tercera clase Sage, Mr. Frederick 9 8 CA. 2343  
## 325 Tercera clase Sage, Mr. George John Jr 9 8 CA. 2343  
## 793 Tercera clase Sage, Miss. Stella Anna 9 8 CA. 2343  
## 847 Tercera clase Sage, Mr. Douglas Bullen 11 8 CA. 2343  
## 864 Tercera clase Sage, Miss. Dorothy Edith "Dolly" 9 8 CA. 2343

En este último caso vemos que se registran 8 personas a bordo, sólo hay 7 pero estamos tratando el data.train de esta competición de kraggle, el otro hermano está en el data.test, por lo que lo damos por bueno

Analizamos ahora los extremos en el número de familiares

unique(boxplot.stats(data.train$Parch)$out)

## [1] 1 2 5 3 4 6

Se registran como valores extremos todo lo que sea diferente a 0. Los pasajeros con 0 hermanos a bordo serán la norma, pero no parece descabellado que haya grupos de hermanos a bordo, por lo quedaremos directamente por buenos los valores diferentes a cero que no sean muy elevados. Como en el caso de los hermanos sólo inspeccionaremos los dos valores más extremos, en este caso 5 y 6.

data.train[data.train$Parch==5,c(3,4,6,7,8,9)]

## Pclass Name Age  
## 14 Tercera clase Andersson, Mr. Anders Johan 39  
## 26 Tercera clase Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson) 38  
## 611 Tercera clase Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren) 39  
## 639 Tercera clase Panula, Mrs. Juha (Maria Emilia Ojala) 41  
## 886 Tercera clase Rice, Mrs. William (Margaret Norton) 39  
## SibSp Parch Ticket  
## 14 1 5 347082  
## 26 1 5 347077  
## 611 1 5 347082  
## 639 0 5 3101295  
## 886 0 5 382652

Si miramos los datos de los tickets de las personas que tienen 5 familiares a bordo podemos detectar si hay alguna anomalía

data.train[data.train$Ticket=="347082",c(3,4,6,7,8,9)]

## Pclass Name Age  
## 14 Tercera clase Andersson, Mr. Anders Johan 39  
## 120 Tercera clase Andersson, Miss. Ellis Anna Maria 2  
## 542 Tercera clase Andersson, Miss. Ingeborg Constanzia 9  
## 543 Tercera clase Andersson, Miss. Sigrid Elisabeth 11  
## 611 Tercera clase Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren) 39  
## 814 Tercera clase Andersson, Miss. Ebba Iris Alfrida 6  
## 851 Tercera clase Andersson, Master. Sigvard Harald Elias 4  
## SibSp Parch Ticket  
## 14 1 5 347082  
## 120 4 2 347082  
## 542 4 2 347082  
## 543 4 2 347082  
## 611 1 5 347082  
## 814 4 2 347082  
## 851 4 2 347082

data.train[data.train$Ticket=="347077",c(3,4,6,7,8,9)]

## Pclass Name Age  
## 26 Tercera clase Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson) 38  
## 183 Tercera clase Asplund, Master. Clarence Gustaf Hugo 9  
## 234 Tercera clase Asplund, Miss. Lillian Gertrud 5  
## 262 Tercera clase Asplund, Master. Edvin Rojj Felix 3  
## SibSp Parch Ticket  
## 26 1 5 347077  
## 183 4 2 347077  
## 234 4 2 347077  
## 262 4 2 347077

data.train[data.train$Ticket=="382652",c(3,4,6,7,8,9)]

## Pclass Name Age SibSp Parch Ticket  
## 17 Tercera clase Rice, Master. Eugene 2 4 1 382652  
## 172 Tercera clase Rice, Master. Arthur 4 4 1 382652  
## 279 Tercera clase Rice, Master. Eric 7 4 1 382652  
## 788 Tercera clase Rice, Master. George Hugh 8 4 1 382652  
## 886 Tercera clase Rice, Mrs. William (Margaret Norton) 39 0 5 382652

data.train[data.train$Ticket=="3101295",c(3,4,6,7,8,9)]

## Pclass Name Age SibSp Parch  
## 51 Tercera clase Panula, Master. Juha Niilo 7 4 1  
## 165 Tercera clase Panula, Master. Eino Viljami 1 4 1  
## 267 Tercera clase Panula, Mr. Ernesti Arvid 16 4 1  
## 639 Tercera clase Panula, Mrs. Juha (Maria Emilia Ojala) 41 0 5  
## 687 Tercera clase Panula, Mr. Jaako Arnold 14 4 1  
## 825 Tercera clase Panula, Master. Urho Abraham 2 4 1  
## Ticket  
## 51 3101295  
## 165 3101295  
## 267 3101295  
## 639 3101295  
## 687 3101295  
## 825 3101295

Parece todo correcto

Miramos el caso de 6 familiares

data.train[data.train$Parch==6,c(3,4,6,7,8,9)]

## Pclass Name Age SibSp Parch  
## 679 Tercera clase Goodwin, Mrs. Frederick (Augusta Tyler) 43 1 6  
## Ticket  
## 679 CA 2144

data.train[data.train$Ticket=="CA 2144",c(3,4,6,7,8,9)]

## Pclass Name Age SibSp Parch  
## 60 Tercera clase Goodwin, Master. William Frederick 11 5 2  
## 72 Tercera clase Goodwin, Miss. Lillian Amy 16 5 2  
## 387 Tercera clase Goodwin, Master. Sidney Leonard 1 5 2  
## 481 Tercera clase Goodwin, Master. Harold Victor 9 5 2  
## 679 Tercera clase Goodwin, Mrs. Frederick (Augusta Tyler) 43 1 6  
## 684 Tercera clase Goodwin, Mr. Charles Edward 14 5 2  
## Ticket  
## 60 CA 2144  
## 72 CA 2144  
## 387 CA 2144  
## 481 CA 2144  
## 679 CA 2144  
## 684 CA 2144

En este caso parece todo correcto, pues faltarían un marido y un hijo que estarán en el data.test.

Para analizar los precios de los tickets lo haremos por clases en lugar de con toda la muestra, pues nos ayudará a identificar mejor las anomalías en esta variable

data.train.firstclass<-data.train[data.train$Pclass=="Primera clase",]  
unique(boxplot.stats(data.train.firstclass$Fare)$out)

## [1] 263.0000 247.5208 512.3292 262.3750 211.5000 227.5250 221.7792 211.3375

Todos los valores detectados están en órdenes de magnitud parecidos, excepto el que supera 500. Miramos este caso, pues los demás son totalmente aceptables

data.train[data.train$Fare>500,c(3,4,6,7,8,9,10)]

## Pclass Name Age SibSp Parch Ticket  
## 259 Primera clase Ward, Miss. Anna 35 0 0 PC 17755  
## 680 Primera clase Cardeza, Mr. Thomas Drake Martinez 36 0 1 PC 17755  
## 738 Primera clase Lesurer, Mr. Gustave J 35 0 0 PC 17755  
## Fare  
## 259 512.3292  
## 680 512.3292  
## 738 512.3292

Tenemos aquí un valor que podría parecer sospechoso, pues pagan por 3 personas más del doble que cualquiera de los otros pasajeros con tickets similares. No obstante, contrastando los nombres de los pasajeros con los datos en Internet está registrado que pagaron 512 $ por sus billetes.

data.train.secondclass<-data.train[data.train$Pclass=="Segunda clase",]  
unique(boxplot.stats(data.train.secondclass$Fare)$out)

## [1] 73.5 65.0

Para la segunda clase parecen del todo razonables los valores detectados como extremos

Analizamos ahora la tercera clase

data.train.thirdclass<-data.train[data.train$Pclass=="Tercera clase",]  
unique(boxplot.stats(data.train.thirdclass$Fare)$out)

## [1] 31.2750 29.1250 31.3875 39.6875 46.9000 27.9000 56.4958 34.3750 69.5500

Aquí llama la atención los valores que superan los 50 dólares, pues serían muy altos incluso para la segunda clase. Miramos si hay muchas personas en el ticket y si no fuera así, deberíamos aplicar alguna corrección o marcarlos como “sospechosos”

data.train.thirdclass[data.train.thirdclass$Fare>50,c(3,4,6,7,8,9,10)]

## Pclass Name Age SibSp Parch Ticket  
## 75 Tercera clase Bing, Mr. Lee 32 0 0 1601  
## 160 Tercera clase Sage, Master. Thomas Henry 9 8 2 CA. 2343  
## 170 Tercera clase Ling, Mr. Lee 28 0 0 1601  
## 181 Tercera clase Sage, Miss. Constance Gladys 9 8 2 CA. 2343  
## 202 Tercera clase Sage, Mr. Frederick 9 8 2 CA. 2343  
## 325 Tercera clase Sage, Mr. George John Jr 9 8 2 CA. 2343  
## 510 Tercera clase Lang, Mr. Fang 26 0 0 1601  
## 644 Tercera clase Foo, Mr. Choong 29 0 0 1601  
## 693 Tercera clase Lam, Mr. Ali 29 0 0 1601  
## 793 Tercera clase Sage, Miss. Stella Anna 9 8 2 CA. 2343  
## 827 Tercera clase Lam, Mr. Len 32 0 0 1601  
## 839 Tercera clase Chip, Mr. Chang 32 0 0 1601  
## 847 Tercera clase Sage, Mr. Douglas Bullen 11 8 2 CA. 2343  
## 864 Tercera clase Sage, Miss. Dorothy Edith "Dolly" 9 8 2 CA. 2343  
## Fare  
## 75 56.4958  
## 160 69.5500  
## 170 56.4958  
## 181 69.5500  
## 202 69.5500  
## 325 69.5500  
## 510 56.4958  
## 644 56.4958  
## 693 56.4958  
## 793 69.5500  
## 827 56.4958  
## 839 56.4958  
## 847 69.5500  
## 864 69.5500

Observamos que estos precios corresponden con dos tickets de 7 personas cada una, por lo que consideramos que los valores son razonables.

# 4. Análisis de los datos

Añadimos dos nuevas variables para facilitar el análisis de la supervivencia, que es la variable en la que centraremos nuestro análisis

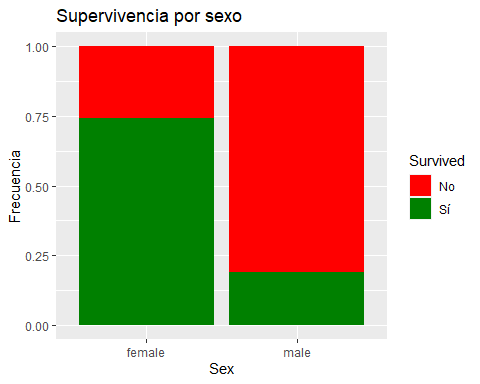
# Añadimos variable FamilyMembers  
data.train$FamilyMembers <- as.integer(data.train$SibSp + data.train$Parch + 1)  
# Añadimos variable FarePerPassenger  
data.train$FarePerPassenger <- data.train$Fare / data.train$FamilyMembers  
# Discretizamos FarePerPassenger  
data.train$FarePerPassengerSegments <- discretize(data.train$FarePerPassenger, method = "interval", breaks = 8)

Exportación de los datos preprocesados

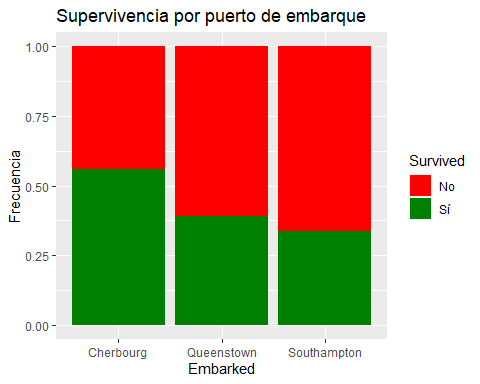
write.csv(data.train, "./titanic/train\_clean.csv")

Analizamos la supervivencia según las otras variables como exploración previa de los datos para seleccionar los grupos a analizar

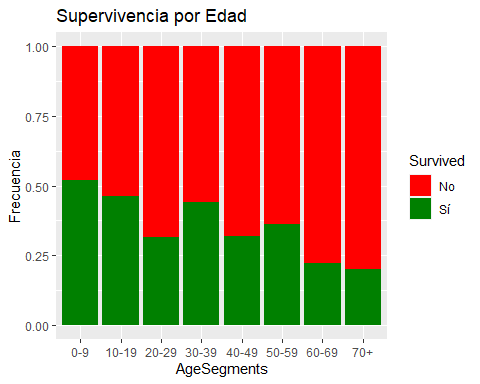
# Survived como función de Sex  
ggplot(data = data.train[1:dim(data.train)[1],],aes(x=Sex,fill=Survived))+geom\_bar(position="fill")+scale\_fill\_manual(values=c("#FF0000","#008000"))+ylab("Frecuencia")+labs(title="Supervivencia por sexo")



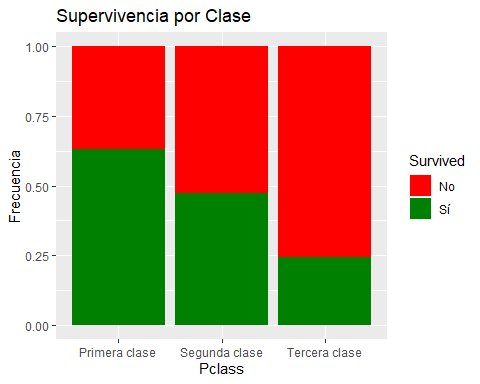
#Survived como función de Embarked:  
ggplot(data = data.train[1:dim(data.train)[1],],aes(x=Embarked,fill=Survived))+geom\_bar(position="fill")+scale\_fill\_manual(values=c("#FF0000","#008000"))+ylab("Frecuencia")+labs(title="Supervivencia por puerto de embarque")



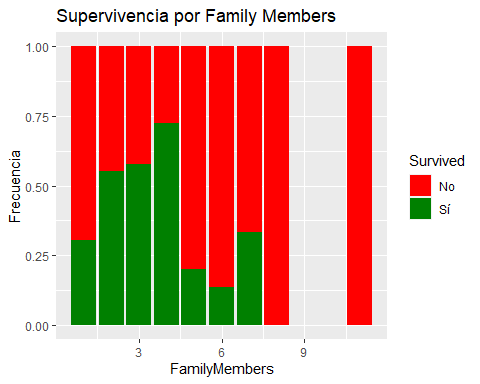
#Survived como función de AgeSegments:  
ggplot(data = data.train[1:dim(data.train)[1],],aes(x=AgeSegments,fill=Survived))+geom\_bar(position="fill")+scale\_fill\_manual(values=c("#FF0000","#008000"))+ylab("Frecuencia")+labs(title="Supervivencia por Edad")



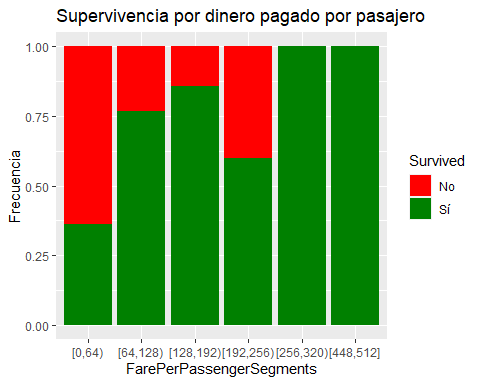
#Survived como función de Pclass:  
ggplot(data = data.train[1:dim(data.train)[1],],aes(x=Pclass,fill=Survived))+geom\_bar(position="fill")+scale\_fill\_manual(values=c("#FF0000","#008000"))+ylab("Frecuencia")+labs(title="Supervivencia por Clase")



#Survived como función de FamilyMembers:  
ggplot(data = data.train[1:dim(data.train)[1],],aes(x=FamilyMembers,fill=Survived))+geom\_bar(position="fill")+scale\_fill\_manual(values=c("#FF0000","#008000"))+ylab("Frecuencia")+labs(title="Supervivencia por Family Members")



#Survived como función de FarePerPassengerSegments:  
ggplot(data = data.train[1:dim(data.train)[1],],aes(x=FarePerPassengerSegments,fill=Survived))+geom\_bar(position="fill")+scale\_fill\_manual(values=c("#FF0000","#008000"))+ylab("Frecuencia")+labs(title="Supervivencia por dinero pagado por pasajero")



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

Tras mirar las distribuciones previas nos puede interesar comparar entre clases, entres sexos y entre puertos de embarque, por lo que creamos los distintos grupos que podemos utilizar

# División en clases  
primera\_clase <- data.train[data.train$Pclass=="Primera clase",]  
segunda\_clase <- data.train[data.train$Pclass=="Segunda clase",]  
tercera\_clase <- data.train[data.train$Pclass=="Tercera clase",]  
# División por supervivencia  
sobrevive <- data.train[data.train$Survived=="Sí",]  
no\_sobrevive <- data.train[data.train$Survived=="No",]  
# División por sexos  
hombre <- data.train[data.train$Sex=="male",]  
mujer <- data.train[data.train$Sex=="female",]  
# División por puerto de embarque  
southampton <- data.train[data.train$Embarked=="Southampton",]  
cherbourg <- data.train[data.train$Embarked=="Cherbourg",]  
queenstown <- data.train[data.train$Embarked=="Queenstown",]

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

El siguiente paso será comprobar que nuestras variables cuantitativas tienen una distribución normal y que sus varianzas son homogéneas. Todas nuestras variables excepto Fare y Age son etiquetas descriptivas, por lo que sólo deberemos hacer los tests de normalidad homocedasticidad a estas dos.

Por el teorema del límite central podríamos asumir normalidad en todos los casos, pues siempre tenemos un número de muestras mayor que 30, pero vamos a asegurarnos. Aplcamis primero el test de Shapiro a ambas variables para comprobar si podemos rechazar la hipótesis nula de que la distribución no es normal con un intervalo de confianza del 95 %

shapiro.test(data.train$Age)

##   
## Shapiro-Wilk normality test  
##   
## data: data.train$Age  
## W = 0.9824, p-value = 7.011e-09

shapiro.test(data.train$Fare)

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

En ambos casos obtenemos un p-valor mucho menos que 0.05, por lo que se comprueba que las distribuciones se asemejan mucho a una normal. Comprobamos ahora la homogeneidad de las varianzas con el test de Levene, que aplicaremos a las edades y precios en función de los grupos seleccionados como de interés. Empezamos con la edad.

leveneTest(data.train$Age~data.train$Pclass)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 2 7.2916 0.0007228 \*\*\*  
## 888   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

leveneTest(data.train$Age~data.train$Sex)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 1 0.7836 0.3763  
## 889

leveneTest(data.train$Age~data.train$Embarked)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 2 0.3627 0.6959  
## 888

leveneTest(data.train$Age~data.train$Survived)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 1 2.608 0.1067  
## 889

Observamos que la edad presenta varianzas distintas según la clase (el p-valor es menor que 0.05 y rechazamos la hipótesis nula). En el caso del sexo, puerto de embarque y supervivencia tendríamos varianzas de edad muy similares para los distintos grupos de cada variable.

leveneTest(data.train$Fare~data.train$Pclass)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 2 118.57 < 2.2e-16 \*\*\*  
## 888   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

leveneTest(data.train$Fare~data.train$Sex)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 1 19.188 1.326e-05 \*\*\*  
## 889   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

leveneTest(data.train$Fare~data.train$Embarked)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 2 33.539 9.082e-15 \*\*\*  
## 888   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

leveneTest(data.train$Fare~data.train$Survived)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 1 45.1 3.337e-11 \*\*\*  
## 889   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Para el caso de los precios observamos que el p-valor es muy pequeño (mucho menor que 0.05) para todas las variables estudiadas, por lo que podemos asumir que las varianzas de precios serán distintas para todos los grupos posibles según la clase, edad, sexo y supervivencia.

## 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.

Vamos a hacer un test de hipotésis sobre la proporción de supervivientes dependiendo de la clase en la que viajaba el pasajero.

Hacemos un test sobre la proporción de supervivientes con 3 muestras, una para cada clase.

p\_sob\_primera <- nrow(primera\_clase[primera\_clase$Survived=="Sí",])/nrow(primera\_clase)  
p\_sob\_segunda <- nrow(segunda\_clase[segunda\_clase$Survived=="Sí",])/nrow(segunda\_clase)  
p\_sob\_tercera <- nrow(tercera\_clase[tercera\_clase$Survived=="Sí",])/nrow(tercera\_clase)  
sob <- c(p\_sob\_primera\*nrow(primera\_clase),p\_sob\_segunda\*nrow(segunda\_clase),p\_sob\_tercera\*nrow(tercera\_clase))  
nn <- c(nrow(primera\_clase),nrow(segunda\_clase),nrow(tercera\_clase))  
prop.test(sob,nn,alternative="two.sided",correct=FALSE)

##   
## 3-sample test for equality of proportions without continuity  
## correction  
##   
## data: sob out of nn  
## X-squared = 102.89, df = 2, p-value < 2.2e-16  
## alternative hypothesis: two.sided  
## sample estimates:  
## prop 1 prop 2 prop 3   
## 0.6296296 0.4728261 0.2423625

El p-valor de 2.2e-16, mucho menor a 0.05, nos hace rechazar la hipótesis nula, por lo que podemos concluir que cada grupo tiene una proporción de supervivientes bastante diferenciada, centrada en los siguientes valores:

Primera clase: 62.96 % de supervivientes

Segunda clase: 47.28 % de supervivientes

Tercera clase: 24.23 % de supervivientes

En el test de correlación vamos a mirar primero las correlaciones entre sobrevivir y alguna variables numéricas

data.train$SurvivedInt <- as.integer(ifelse(data.train$Survived=="Sí",1,0))  
cor.test(data.train$FarePerPassenger,data.train$SurvivedInt)

##   
## Pearson's product-moment correlation  
##   
## data: data.train$FarePerPassenger and data.train$SurvivedInt  
## t = 6.7757, df = 889, p-value = 2.251e-11  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.1582253 0.2831562  
## sample estimates:  
## cor   
## 0.2215999

cor.test(data.train$Age,data.train$SurvivedInt)

##   
## Pearson's product-moment correlation  
##   
## data: data.train$Age and data.train$SurvivedInt  
## t = -2.9522, df = 889, p-value = 0.003238  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.16315413 -0.03306915  
## sample estimates:  
## cor   
## -0.09853255

cor.test(data.train$SibSp,data.train$SurvivedInt)

##   
## Pearson's product-moment correlation  
##   
## data: data.train$SibSp and data.train$SurvivedInt  
## t = -1.0538, df = 889, p-value = 0.2922  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.10076614 0.03042549  
## sample estimates:  
## cor   
## -0.0353225

Observamos cómo la correlación entre la edad, el número de hermanos o cónyugues y la supervivencia es prácticamente inexistente. En estos casos los p-valores son altos y la correlación bastante baja.

No obstante, para la dependencia de la supervivencia con el precio del billete por pasajero sí que hay un p-valor muy pequeño que permite rechazar la hipótesis nula y la correlación es de 0.22. Esta correlación no es excesivamente grande, pero teniendo en cuenta que sobrevivir es una variable binaria que da 0 o 1, una correlación de 0.22 será muy a tener en cuenta.

Para finalizar vamos a usar una regresión logística para predecir la probabilidad de supervivencia en función de las variables que hemos encontrado que puedan tener algún efecto en la misma. Nos decantamos por usar el sexo, el puerto de embarque, la clase y el precio.

glm\_sobrevivir <- glm(formula =Survived~Sex+Embarked+Pclass+FarePerPassenger, family=binomial(link=logit),data=data.train)  
summary(glm\_sobrevivir)

##   
## Call:  
## glm(formula = Survived ~ Sex + Embarked + Pclass + FarePerPassenger,   
## family = binomial(link = logit), data = data.train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2914 -0.7140 -0.4181 0.6707 2.2394   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.442059 0.322893 7.563 3.94e-14 \*\*\*  
## Sexmale -2.604784 0.185739 -14.024 < 2e-16 \*\*\*  
## EmbarkedQueenstown -0.113224 0.363962 -0.311 0.7557   
## EmbarkedSouthampton -0.557462 0.231102 -2.412 0.0159 \*   
## PclassSegunda clase -0.556207 0.273310 -2.035 0.0418 \*   
## PclassTercera clase -1.702389 0.257509 -6.611 3.82e-11 \*\*\*  
## FarePerPassenger 0.003764 0.003503 1.075 0.2826   
## ---  
## 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: 817.44 on 884 degrees of freedom  
## AIC: 831.44  
##   
## Number of Fisher Scoring iterations: 5

Concluimos de este modelo que las variables que más influyen en la no supervivencia son ser hombre y ser de tercera clase, pues tenemos unos p-valores muy pequeños para ambas variables en el modelo.

Creado el modelo miramos la tabla de confusión de las predicciones hechas con el mismo y lo comparamos con las personas que han sobrevivido

library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

library(e1071)

## Warning: package 'e1071' was built under R version 4.0.3

predicted\_data <- predict(glm\_sobrevivir, data.train)  
predicted\_survival <- as.factor(ifelse(predicted\_data>0.5,"Sí","No"))  
cm <- confusionMatrix(predicted\_survival,data.train$Survived)  
cm$table

## Reference  
## Prediction No Sí  
## No 522 140  
## Sí 27 202

La exactitud del modelo es del 81.33 %. 523 valores son muertos reales y 200 son supervivientes reales del total de 891 resultados totales. El 81.33 % de las predicciones son correctas.

La precisión es elevada, del 88.49 %. El modelo predice bien las personas que sobreviven: 200 casos de supervivencia son correctos de los 226 que predecimos.

La especificidad es muy buena, del 95.26%. Esto quiere decir que el modelo predice muy bien las personas que NO van a sobrevivir, prediciendo 523 personas que mueren de las 549 que en realidad murieron.

El peor de los parámetros de predicción es la sensibilidad, ya que de las 340 personas que en realidad sobreviven, tan sólo se detectan correctamente 200, un 58.82 %.

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

Representación de la curva ROC

library(pROC)

## Warning: package 'pROC' was built under R version 4.0.3

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following object is masked from 'package:colorspace':  
##   
## coords

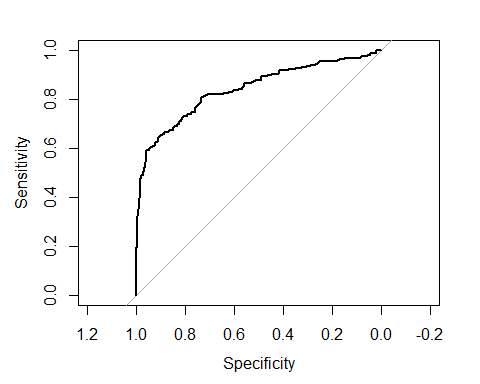
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

r=roc(data.train$Survived,predicted\_data , data=data.train)

## Setting levels: control = No, case = Sí

## Setting direction: controls < cases

plot(r)



auc(r)

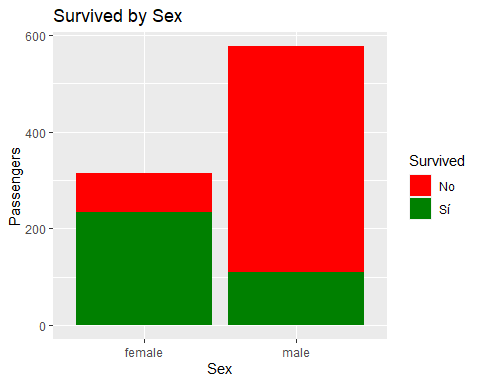
## Area under the curve: 0.8399

El área por debajo de esa curva toma el valor de 0.8397, por lo que la habilidad del modelo para discriminar entre aquellos pasajeros que sobrevivieron y los que no, es buena.

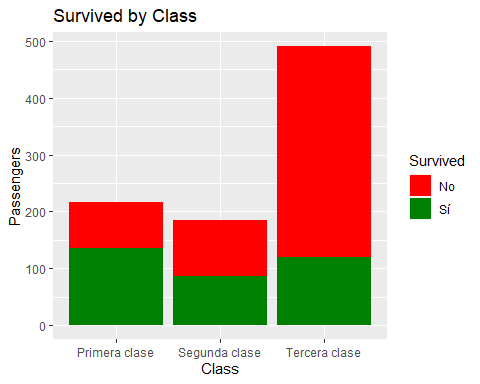
Representamos gráficamente la relación entre la supervivencia y cada uno de las siguientes variables el sexo y la clase en la que viajaban, mediante diagramas de barras la cantidad de supervivientes y no supervivientes

Podemos comprobar gráficamente la conclusión que extrajimos, las variables que más influyen en la no supervivencia son ser hombre y viajar en tercera clase

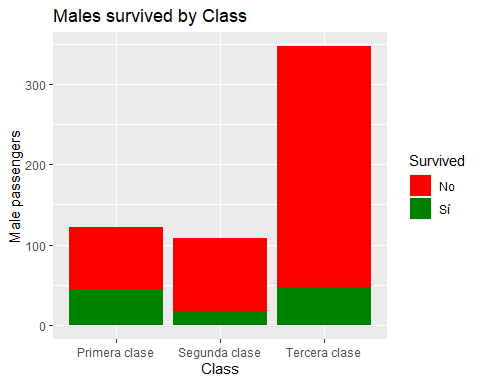
ggplot(data.train,aes(Sex ,fill=Survived))+geom\_bar() +labs(x="Sex", y="Passengers")+  
scale\_fill\_manual(values=c("#FF0000","#008000"))+ggtitle("Survived by Sex")



ggplot(data.train,aes(Pclass ,fill=Survived))+geom\_bar() +labs(x="Class", y="Passengers")+  
scale\_fill\_manual(values=c("#FF0000","#008000"))+ggtitle("Survived by Class")



ggplot(hombre,aes(Pclass ,fill=Survived))+geom\_bar() +labs(x="Class", y="Male passengers")+  
scale\_fill\_manual(values=c("#FF0000","#008000"))+ggtitle("Males survived by Class")



Tablas de contingencia

SurvivedSex <- table(data.train$Sex, data.train$Survived)  
SurvivedSex

##   
## No Sí  
## female 81 233  
## male 468 109

prop.table(SurvivedSex, margin = 1)

##   
## No Sí  
## female 0.2579618 0.7420382  
## male 0.8110919 0.1889081

SurvivedClass <- table(data.train$Pclass,data.train$Survived)  
SurvivedClass

##   
## No Sí  
## Primera clase 80 136  
## Segunda clase 97 87  
## Tercera clase 372 119

prop.table(SurvivedClass, margin = 1)

##   
## No Sí  
## Primera clase 0.3703704 0.6296296  
## Segunda clase 0.5271739 0.4728261  
## Tercera clase 0.7576375 0.2423625

SurvivedSexbyClass <- table(data.train$Sex,data.train$Survived,data.train$Pclass)  
SurvivedSexbyClass

## , , = Primera clase  
##   
##   
## No Sí  
## female 3 91  
## male 77 45  
##   
## , , = Segunda clase  
##   
##   
## No Sí  
## female 6 70  
## male 91 17  
##   
## , , = Tercera clase  
##   
##   
## No Sí  
## female 72 72  
## male 300 47

prop.table(SurvivedSexbyClass, margin = 1)

## , , = Primera clase  
##   
##   
## No Sí  
## female 0.00955414 0.28980892  
## male 0.13344887 0.07798960  
##   
## , , = Segunda clase  
##   
##   
## No Sí  
## female 0.01910828 0.22292994  
## male 0.15771231 0.02946274  
##   
## , , = Tercera clase  
##   
##   
## No Sí  
## female 0.22929936 0.22929936  
## male 0.51993068 0.08145581

# 6. Resolución del problema. Conclusiones.

Para la resolución realizamos las siguientes acciones y extrajimos las correspondientes conclusiones

#### Realizamos un test sobre la proporción de supervivientes con 3 muestras, una para cada clase, y concluimos que cada grupo tiene una proporción de supervivientes bastante diferenciada, centrada en los siguientes valores:

Primera clase: 62.96 % de supervivientes

Segunda clase: 47.28 % de supervivientes

Tercera clase: 24.23 % de supervivientes

#### Realizamos test de correlación entre sobrevivir y las variables numéricas FarePerPassenger, Age y SibSp, y concluímos que:

La correlación entre la edad, el número de hermanos o cónyugues y la supervivencia es prácticamente inexistente.

La correlación del precio del billete por pasajero y la supervivencia no es excesivamente grande, pero teniendo en cuenta que sobrevivir es una variable binaria que da 0 o 1, una correlación de 0.22 será muy a tener en cuenta

#### Realizamos una regresión logística para predecir la probabilidad de supervivencia en función de las variables que hemos encontrado que puedan tener algún efecto en la misma. Nos decantamos por usar el sexo, el puerto de embarque, la clase y el precio. Concluímos que:

La exactitud del modelo es del 81.33 %. 523 valores son muertos reales y 200 son supervivientes reales del total de 891 resultados totales. El 81.33 % de las predicciones son correctas.

La precisión es elevada, del 88.49 %. El modelo predice bien las personas que sobreviven: 200 casos de supervivencia son correctos de los 226 que predecimos.

La especificidad es muy buena, del 95.26%. Esto quiere decir que el modelo predice muy bien las personas que NO van a sobrevivir, prediciendo 523 personas que mueren de las 549 que en realidad murieron.

El peor de los parámetros de predicción es la sensibilidad, ya que de las 340 personas que en realidad sobreviven, tan sólo se detectan correctamente 200, un 58.82 %.