AI2

Cleber Perez

2024-11-20

Bibliotecas

```
# Cargamos todas las librería en la lista "librerias"
librerias =
c('tidyverse','broom','ISLR','GGally','modelr','cowplot','rlang','modelr','ti
bble','Metrics','mice','visdat',"caret")

for (lib in librerias){
  library(lib,character.only=TRUE)}
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
##
##
## Attaching package: 'modelr'
##
##
## The following object is masked from 'package:broom':
##
##     bootstrap
##
##
##
## Attaching package: 'cowplot'
##
##
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
##
##
##
## Attaching package: 'rlang'
##
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
##
##
##
## Attaching package: 'Metrics'
##
##
## The following object is masked from 'package:rlang':
##
##     ll
##
##
## The following objects are masked from 'package:modelr':
##
##     mae, mape, mse, rmse
##
##
##
## Attaching package: 'mice'
##
##
## The following object is masked from 'package:stats':
##
##     filter
##
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
##
##
## Loading required package: lattice
##
##
## Attaching package: 'caret'
##
##
## The following objects are masked from 'package:Metrics':
```

```
##
##     precision, recall
##
##
## The following object is masked from 'package:purrr':
##
##     lift
```

Leyendo los datos:

```
M = read.csv("Titanic.csv")
str(M)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Survived   : int  0 1 0 0 1 0 1 0 1 0 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)"
"Myles, Mr. Thomas Francis" "Wirz, Mr. Albert" ...
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Las variables son:

- *Name:* Nombre del pasajero
- *PassengerId:* Ids del pasajero
- *Survived:* Si sobrevivió o no (No = 0, Sí = 1)
- *Ticket:* Número de ticket
- *Cabin:* Cabina en la que viajó
- *Pclass:* Clase en la que viajó (1 = 1era, 2 = 2da, 3 = 3ra)
- *Sex:* Masculino o Femenino (male/female)
- *Age:* Edad

- *SibSp:* Número de hermanos/conyuge a bordo
- *Parch:* Número de padres/hijos a bordo
- *Fare:* Tarifa que pagó
- *Embarked:* Puerto de embarcación (C = Cherbourg, Q = Queenstown, S = Southampton)

Preparación de la base de datos

Ajustando las variables

Variables de interés: Quita aquellas que de entrada no tengan que ver con la sobrevivencia del pasajero. Por ejemplo: Quitar variables 4, 9 y 11 (define si hay más)

Tambien quite la variable 10 debido a que esta es solo la tarifa que se pago, pero ya tenemos una variable que recupera en que clase se viajo.

Variables categóricas que deben aparecer como factores: define qué variables aparecerán como factores Por ejemplo: Survived, Pclass, Sex y Embarked (define si hay más)

```
# Eliminar variables:
M1 <- M[,c(-4,-9,-10,-11)]

#Transformar a factores:
for(var in c('Survived','Pclass','Embarked','Sex'))
  M1[,var] <-as.factor(M1[,var])
```

Análisis de datos faltantes

Detectar si hay espacios vacíos en lugar de datos:

```
V = matrix(NA,ncol=1,nrow=8)
for(i in c(1:8)){
  V[i,] <- sum(with(M1,M1[,i])=="")}
V
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
## [4,]    0
## [5,]   NA
## [6,]    0
## [7,]    0
## [8,]   NA
```

Ninguna variable contiene espacios vacíos, pero las variables 5 (Age) y 8 (Embarked) tienen datos faltantes.

Para contar los datos faltantes:

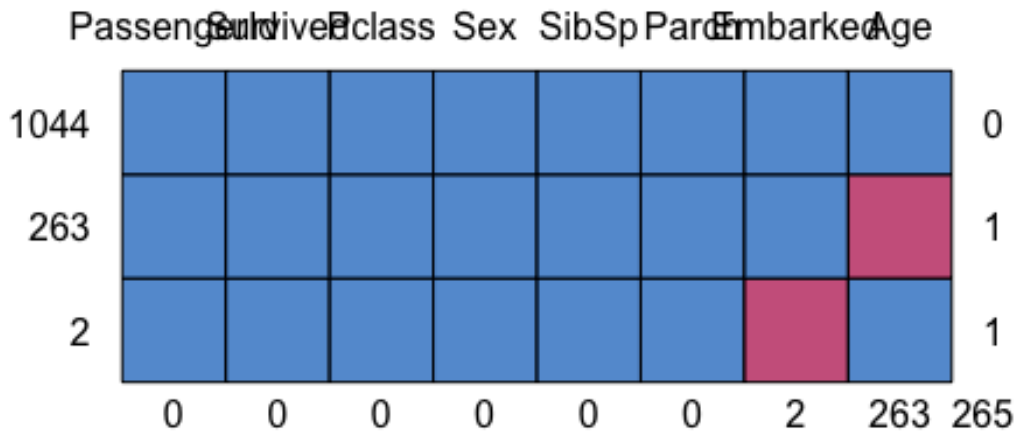
```
N = apply(X=is.na(M1),MARGIN = 2,FUN = sum)
P = round(100*N/length(M1[,2]),2)
NP = data.frame(as.numeric(N),as.numeric(P))
row.names(NP)= c("PassengerId", "Survived", "Pclass", "Sex", "Age", "SibSp",
"Parch", "Embarked")
names(NP)=c("Número","Porcentaje")
t(NP)
```

```
##            PassengerId Survived Pclass Sex    Age SibSp Parch Embarked
## Número               0        0      0   0 263.00     0     0     2.00
## Porcentaje           0        0      0   0  20.09     0     0     0.15
```

En edad hay muchos datos faltantes, el 20% de los datos.

Observemos el patrón de los datos faltantes:

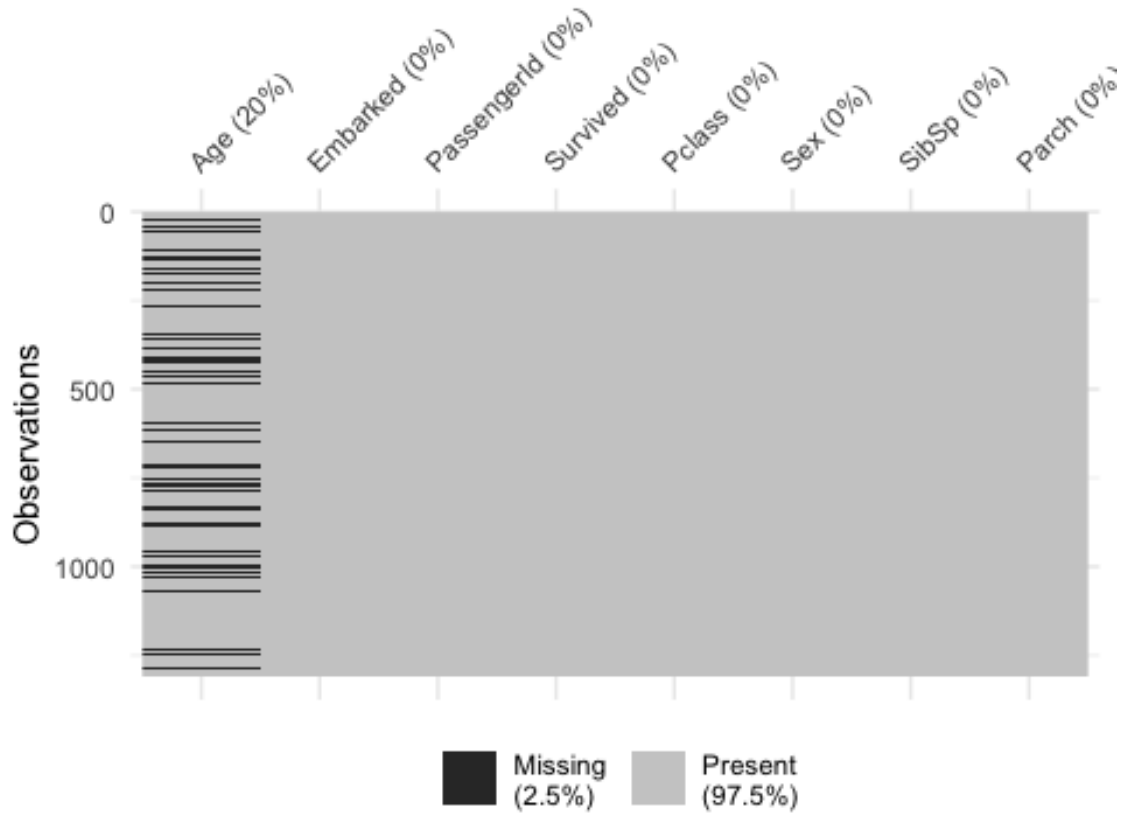
```
md.pattern(M1)
```



```
##      PassengerId Survived Pclass Sex SibSp Parch Embarked Age
## 1044           1        1      1   1     1     1        1   1   0
## 263            1        1      1   1     1     1        1   0   1
## 2              1        1      1   1     1     1        0   1   1
##                0        0      0   0     0     0        2 263 265
```

Todos los datos faltantes son de distintos pasajeros (observaciones), por lo tanto, si se eliminan los NA, se eliminarían 266 observaciones y nos quedaríamos con 1043 observaciones.

```
vis_miss(M1,sort_miss = TRUE)
```



Análisis sobre datos faltantes

Medidas con datos faltantes

```
summary(M1[,-1])
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:815    1:323   female:466   Min.   : 0.17   Min.   :0.0000   Min.
:0.000
##  1:494    2:277   male  :843   1st Qu.:21.00   1st Qu.:0.0000   1st
Qu.:0.000
##           3:709                 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
##  Embarked
##  C   :270
##  Q   :123
##  S   :914
##  NA's:  2
##
##
##
```

Medidas sin datos faltantes

```
M2 = na.omit(M1)
summary(M2[,-1])
```

```
##  Survived Pclass      Sex           Age            SibSp
##  0:629    1:282   female:386   Min.   : 0.17   Min.   :0.0000
##  1:415    2:261   male  :658   1st Qu.:21.00   1st Qu.:0.0000
##           3:501                 Median :28.00   Median :0.0000
##                                 Mean   :29.84   Mean   :0.5038
##                                 3rd Qu.:39.00   3rd Qu.:1.0000
##                                 Max.   :80.00   Max.   :8.0000
##      Parch        Embarked
##  Min.   :0.0000   C:212
##  1st Qu.:0.0000   Q: 50
##  Median :0.0000   S:782
##  Mean   :0.4215
##  3rd Qu.:1.0000
##  Max.   :6.0000
```

¿Difieren las medidas con o sin datos faltantes? ¿cuáles son las variables que más se ven afectadas?

Sobrevivientes

```
t2c = 100*prop.table(table(M1[,2]))
t2s = 100*prop.table(table(M2[,2]))
t2p = c(t2s[1]/t2c[1],t2s[2]/t2c[2])
t2 = data.frame(as.numeric(t2c),as.numeric(t2s),as.numeric(t2p))
row.names(t2) = c("Murió","Sobrevivió")
names(t2) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t2,2)
```

```
##            Con NA (%) Sin NA (%) Pérdida (prop)
## Murió           62.26      60.25           0.97
## Sobrevivió      37.74      39.75           1.05
```

Clase en que viajó

```
t3c = 100*prop.table(table(M1[,3]))
t3s = 100*prop.table(table(M2[,3]))
t3p = c(t3s[1]/t3c[1],t3s[2]/t3c[2],t3s[3]/t3c[3])
t3 = data.frame(as.numeric(t3c),as.numeric(t3s),as.numeric(t3p))
row.names(t3) = c("Primera","Segunda","Tercera")
names(t3) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t3,2)
```

```
##         Con NA (%) Sin NA (%) Pérdida (prop)
## Primera      24.68      27.01           1.09
## Segunda      21.16      25.00           1.18
## Tercera      54.16      47.99           0.89
```

Sexo

```
t4c = 100*prop.table(table(M1[,4]))
t4s = 100*prop.table(table(M2[,4]))
t4p = c(t4s[1]/t4c[1],t4s[2]/t4c[2])
t4 = data.frame(as.numeric(t4c),as.numeric(t4s),as.numeric(t4p))
row.names(t4) = c("Mujer","Hombre")
names(t4) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t4,2)
```

```
##        Con NA (%) Sin NA (%) Pérdida (prop)
## Mujer        35.6      36.97           1.04
## Hombre       64.4      63.03           0.98
```

Puerto de embarcación

```
t8c = 100*prop.table(table(M1[,8]))
t8s = 100*prop.table(table(M2[,8]))
t8p = c(t8s[1]/t8c[1],t8s[2]/t8c[2],t8s[3]/t8c[3])
t8 = data.frame(as.numeric(t8c),as.numeric(t8s),as.numeric(t8p))
row.names(t8) = c("Cherbourg","Queenstown","Southampton")
names(t8) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t8,2)
```

```
##             Con NA (%) Sin NA (%) Pérdida (prop)
## Cherbourg        20.66      20.31           0.98
## Queenstown        9.41       4.79           0.51
## Southampton      69.93      74.90           1.07
```

En este ensayo quitarás los datos faltantes, pero deberás indicar cuáles son las variables más afectadas y por qué.

Donde mas cambio hubo fue en la variable de Clase en la que viajo y en Embarked, se puede observar que gente que tenia variables con NA varios pertenecian a la Tercera clase que es donde mas cambio hubo, y dentro de la variable de Embarked, se puede observar que la variable de Queenstown sin Na tiene una reduccion de 5 unidades.

Análisis descriptivo

Se recomienda analizar dividiendo la base de datos entre los que sobrevivieron y los que no. Usa:

- Medidas
- Gráficos

Partición. Entrenamiento y prueba

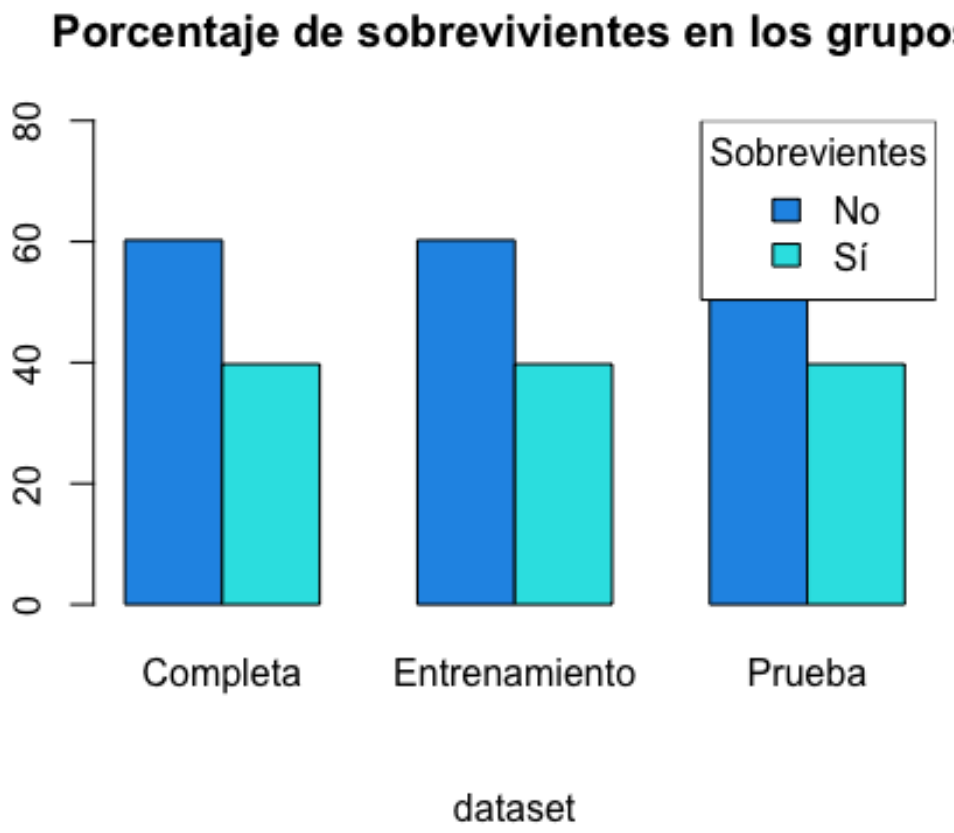
Se toma el 70% de la muestra como entrenamiento y el 30% para prueba.

```
M_indice <- createDataPartition(M2$Survived, p = .7, list = FALSE, times = 1)

M_train <- M2[ M_indice,]  %>% as_tibble()
M_valid <- M2[-M_indice,] %>% as_tibble()
```

Proporciones de sobrevivientes en las tres bases de datos

- Calcula la proporción de sobrevivientes en cada base de datos: Entrenamiento, prueba y completa. Haz una tabla comparativa
- Haz un gráfico de barras que te ayude a comparar las tres bases de datos. Auxíliate del código:

```
prop_completa <- 100 * prop.table(table(M2$Survived))
prop_train <- 100 * prop.table(table(M_train$Survived))
prop_valid <- 100 * prop.table(table(M_valid$Survived))

TablaComparativa <- data.frame(
  Completa = as.numeric(prop_completa),
  Entrenamiento = as.numeric(prop_train),
  Prueba = as.numeric(prop_valid)
)

barplot(as.matrix(TablaComparativa), col=4:5, beside=TRUE, main="Porcentaje
de sobrevivientes en los grupos", sub="dataset",ylim=c(0,80))
legend("topright",legend = c("No","Sí"), title = "Sobrevientes",fill = 4:5)
```



Define si la proporción de no sobrevientes se mantiene en las tres bases de datos.

Si se mantiene

Modelación (entrenamiento)

Comienza con el modelo completo, incluyendo las variables categóricas (factores). Aplica el comando *step* para poder encontrar el mejor modelo.

step utiliza el criterio de Aikaike (AIC) para definir el mejor modelo, sin embargo también proporciona la desviación residual del modelo completo. Un menor AIC y una menor *Deviance* indicarán un mejor modelo.

```
A = glm(Survived ~ ., data = M_train, family = "binomial")

step(A, direction="both", trace=1 )

## Start:  AIC=574.69
## Survived ~ PassengerId + Pclass + Sex + Age + SibSp + Parch +
##     Embarked
##
##               Df Deviance    AIC
## - Embarked     2   557.28 573.28
```

```
## - Parch        1   556.46 574.46
## <none>             554.69 574.69
## - PassengerId  1   557.15 575.15
## - SibSp        1   557.85 575.85
## - Age          1   571.15 589.15
## - Pclass       2   602.80 618.80
## - Sex          1   869.07 887.07
##
## Step:  AIC=573.28
## Survived ~ PassengerId + Pclass + Sex + Age + SibSp + Parch
##
##               Df Deviance    AIC
## - Parch        1   559.14 573.14
## <none>             557.28 573.28
## - PassengerId  1   559.47 573.47
## + Embarked     2   554.69 574.69
## - SibSp        1   560.83 574.83
## - Age          1   575.57 589.57
## - Pclass       2   618.90 630.90
## - Sex          1   876.66 890.66
##
## Step:  AIC=573.14
## Survived ~ PassengerId + Pclass + Sex + Age + SibSp
##
##               Df Deviance    AIC
## <none>             559.14 573.14
## - PassengerId  1   561.25 573.25
## + Parch        1   557.28 573.28
## + Embarked     2   556.46 574.46
## - SibSp        1   565.00 577.00
## - Age          1   577.12 589.12
## - Pclass       2   622.91 632.91
## - Sex          1   881.93 893.93

##
## Call:  glm(formula = Survived ~ PassengerId + Pclass + Sex + Age + SibSp,
##     family = "binomial", data = M_train)
##
## Coefficients:
## (Intercept)  PassengerId      Pclass2      Pclass3      Sexmale
Age
##   4.5314838   -0.0004143   -1.2115258   -2.2916888   -3.5193906   -
0.0366500
##       SibSp
##  -0.3061696
##
## Degrees of Freedom: 731 Total (i.e. Null);  725 Residual
## Null Deviance:       983.8
## Residual Deviance: 559.1     AIC: 573.1
```

- Identifica el mejor modelo de acuerdo con el AIC
- Selecciona la última variable que elminó el comando *step*. Prueba dos modelos, uno con esa variable y otro sin ella.

Modelo B

- Prueba el modelo incluyendo la última variable que eliminó el comando *step*.
- Indica cuáles son las variables que incluye.
- Interpreta la significancia global (de todo el modelo) y la individual (de cada una de las variables)

```
B = glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch, family =
"binomial", data = M_train)
summary(B)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch,
##     family = "binomial", data = M_train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.31155    0.48545   8.882  < 2e-16 ***
## Pclass2     -1.18045    0.30258  -3.901 9.57e-05 ***
## Pclass3     -2.22815    0.30528  -7.299 2.91e-13 ***
## Sexmale     -3.56930    0.24367 -14.648  < 2e-16 ***
## Age         -0.03708    0.00896  -4.139 3.49e-05 ***
## SibSp       -0.23670    0.13509  -1.752   0.0797 .
## Parch       -0.17500    0.13307  -1.315   0.1885
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 983.81  on 731  degrees of freedom
## Residual deviance: 559.47  on 725  degrees of freedom
## AIC: 573.47
##
## Number of Fisher Scoring iterations: 5
```

Modelo C

- Prueba el modelo tal como te lo recomendó el comando *step*.
- Indica cuáles son las variables que incluye.
- Interpreta la significancia global (de todo el modelo) y la individual (de cada una de las variables)

```
C = glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = "binomial",
data = M_train)
summary(C)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = "binomial",
##     data = M_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.222577   0.478664   8.822  < 2e-16 ***
## Pclass2     -1.195676   0.302361  -3.954 7.67e-05 ***
## Pclass3     -2.256619   0.304769  -7.404 1.32e-13 ***
## Sexmale     -3.503346   0.236462 -14.816  < 2e-16 ***
## Age         -0.036591   0.008912  -4.106 4.03e-05 ***
## SibSp       -0.290188   0.129026  -2.249   0.0245 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 983.81  on 731  degrees of freedom
## Residual deviance: 561.25  on 726  degrees of freedom
## AIC: 573.25
##
## Number of Fisher Scoring iterations: 5
```

Análisis de los modelos B y C

Resumen de los indicadores importantes de los modelos B y C

Compara el AIC, la *Null Deviance* y la *Residual Deviance* de los modelos B y C. Extrae los valores con los modelos con los comandos:

- B$aic
- B$deviance
- B$null.deviance

Elabora una tabla comparativa

```
B_AIC <- B$aic
B_Residual_Deviance <- B$deviance
B_Null_Deviance <- B$null.deviance

C_AIC <- C$aic
C_Residual_Deviance <- C$deviance
C_Null_Deviance <- C$null.deviance

comparativa <- data.frame(
  Indicador = c("AIC", "Residual Deviance", "Null Deviance"),
  Modelo_B = c(B_AIC, B_Residual_Deviance, B_Null_Deviance),
  Modelo_C = c(C_AIC, C_Residual_Deviance, C_Null_Deviance)
)
```

```r
print(comparativa)
```

```
##           Indicador Modelo_B Modelo_C
## 1               AIC 573.4731 573.2485
## 2 Residual Deviance 559.4731 561.2485
## 3     Null Deviance 983.8110 983.8110
```

¿Cómo se comporta la *Null Deviance*? ¿por qué?

La NullDeviance es igual en ambos modelos porque mide la variabilidad de la variable dependiente que en este caso es survived, sin incluir alguna otra variable.

¿Qué pasa con el AIC y la *Residual Deviance*?

El AIC resulta mejor para el modelo C, aunque el Residual Deviance es algo menor en el otro modelo, esto indica que se tiene un ajuste mejor al incluir una variable mas pero esta no resulta altamente significativa debida a su varianza en AIC en ambos modelos.

Cálculo de la Desviación explicada ($pseudor^2$)

Calcula la desviación explicada para cada modelo. Recuerda que es igual a:

pseudo r^2 = 1-Desviación residual/Desviación nula

Compara los resultados obtenidos por ambos modelos

```r
pseudo_r2_B <- 1 - (B$deviance / B$null.deviance)
pseudo_r2_C <- 1 - (C$deviance / C$null.deviance)

cat("Pseudo r^2 para el Modelo B: ", pseudo_r2_B, "\n")
```

```
## Pseudo r^2 para el Modelo B:  0.4313205
```

```r
cat("Pseudo r^2 para el Modelo C: ", pseudo_r2_C, "\n")
```

```
## Pseudo r^2 para el Modelo C:  0.4295159
```

Ambos modelos tienen una Pseudo r^2 muy parecida, aunque el Modelo B tiene una ligeramente mayor.

Prueba de razón de verosimilitud

H_0: El modelo con predictores explica mejor la variable respuesta: $log\left(\frac{p}{1-p}\right)$ que el modelo nulo

H_1: El modelo nulo explica mejor la variable respuesta: $log\left(\frac{p}{1-p}\right)$ (la probabilidad es constante)

Se calcula el estadístico de χ^2 para la razón de verosimilutud a partir de las *Deviance* de los modelos.

```r
n_observaciones <- nrow(M_train)
n_parametros <- length(coef(B))

df_deviance <- n_observaciones - n_parametros
df_deviance
```

```
## [1] 725
```

```r
Diferencia = B$null.deviance-B$deviance
gl = B$df.null - df_deviance

pchisq(Diferencia,gl,lower.tail = FALSE)
```

```
## [1] 1.631656e-88
```

Interpreta en el contexto del problema

Comparación entre los modelos B y C

Se pueden comparar los modelo B y C para ver si hay una diferencia significativa entre ambos con la misma razón de verosimilitud utilizando el comando ANOVA y la prueba LR.

```r
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```r
anova(B,C,test="LR")
```

```
## Analysis of Deviance Table
##
## Model 1: Survived ~ Pclass + Sex + Age + SibSp + Parch
## Model 2: Survived ~ Pclass + Sex + Age + SibSp
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       725     559.47
## 2       726     561.25 -1  -1.7755   0.1827
```

Modelo Seleccionado

Define los coeficientes del modelo seleccionado. Por ejemplo, si el modelo seleccionado fue el C:

```
b0 = round(C$coefficients[1], 3)
b1 = round(C$coefficients[2], 3)
b2 = round(C$coefficients[3], 3)
b3 = round(C$coefficients[4], 3)
b4 = round(C$coefficients[5], 3)
b5 = round(C$coefficients[6], 3)
```

Gráfica el modelo

Para percibir el efecto de cada variable, grafica cada variable contra los valores predichos por el modelo. Aunque en el modelo, la variable respuesta es:

$$\hat{y} = log\left(\frac{p}{1-p}\right)$$

con el subcomando: *fitted.values* del comando *glm* se obtienen las probabilidades estimadas para los valores datos. R despeja las probabilidades:

$$\hat{p} = \left(\frac{e^{\hat{y}}}{1+e^{\hat{y}}}\right)$$

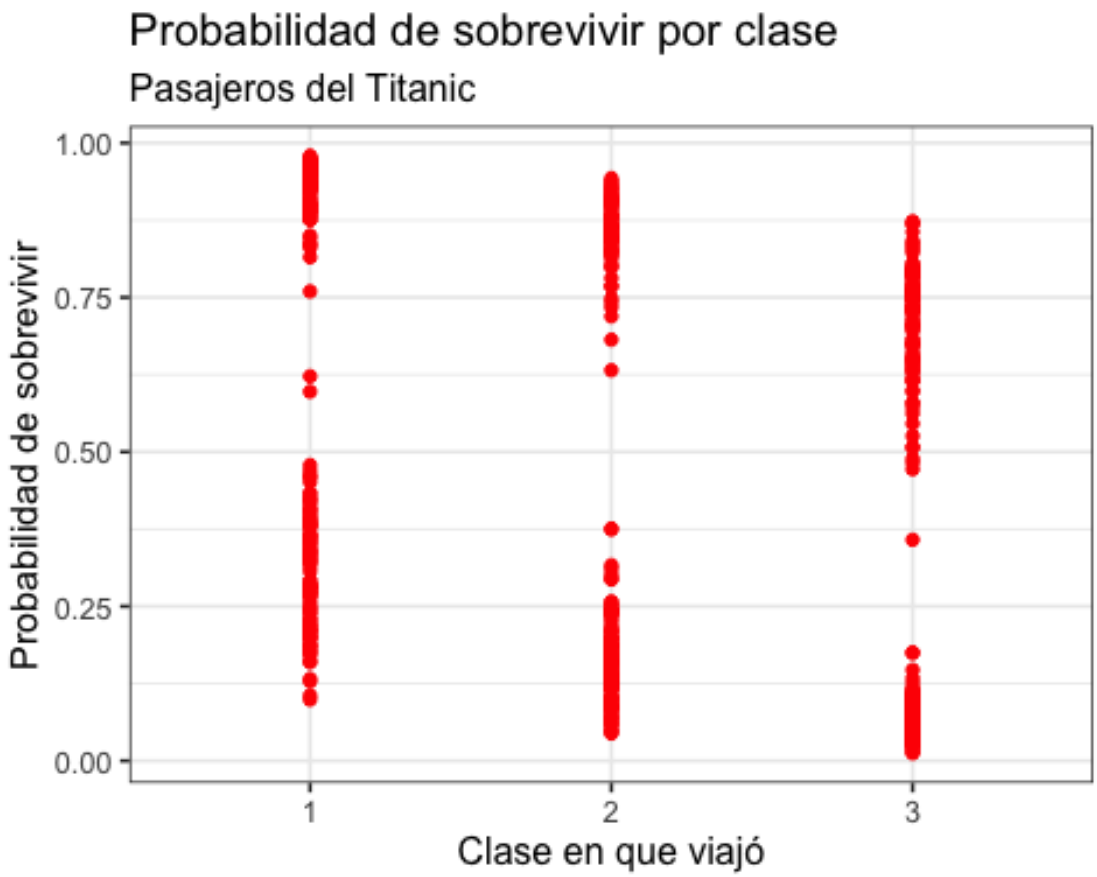
Así que interpretar el efecto de cada variable, se grafica cada una de ellas contra los valores predichos para la probabilidad de sobrevivencia.

Para hacer los gráficos se ejemplifica con:

Clase en que viajó el pasajero

```
p_pred = C$fitted.values
M_pred = data.frame(M_train[,c(2,3,4,5,6)],p_pred)

ggplot(M_pred, aes( x = Pclass)) +
geom_point(aes(y=M_pred$p_pred), size=1.5,color="red") +
  labs(x="Clase en que viajó", y="Probabilidad de sobrevivir",
       title="Probabilidad de sobrevivir por clase",
       subtitle="Pasajeros del Titanic",
       col="")+
  theme_bw(base_size = 12)
```

```
## Warning: Use of `M_pred$p_pred` is discouraged.
## ℹ Use `p_pred` instead.
```



Grafica y concluye cómo cambia la probabilidad predicha con cada variable que resultó significativa

Predicciones

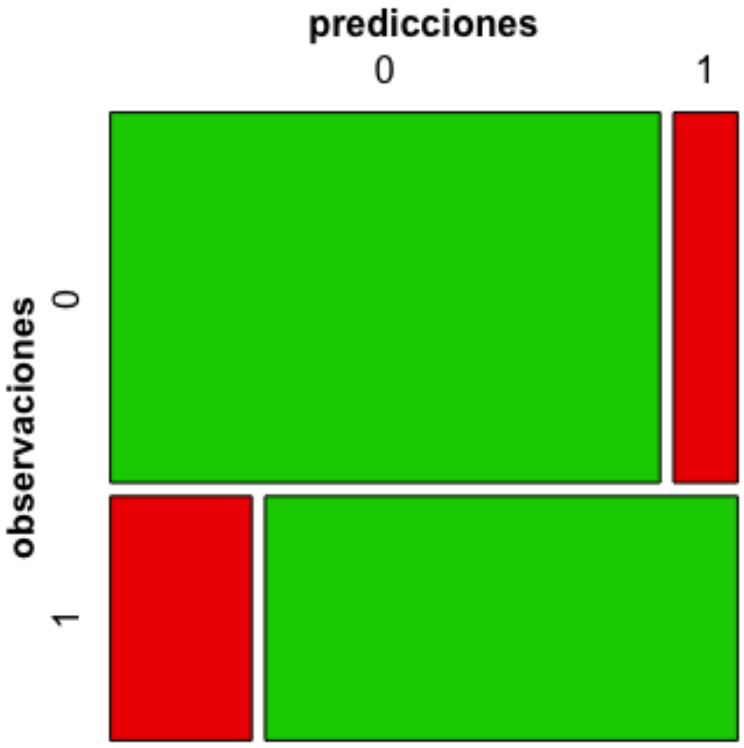
Se hace el análisis con el modelo seleccionado, en el ejemplo suponemos que se seleccionó el modelo C.

Matriz de confusión

```
library(vcd)

## Loading required package: grid

##
## Attaching package: 'vcd'

## The following object is masked from 'package:ISLR':
##
##     Hitters

predicciones <- ifelse(test = C$fitted.values > 0.5, yes = 1, no = 0)
M_C <- table(C$model$Survived, predicciones, dnn = c("observaciones",
"predicciones"))
M_C
```

```
##              predicciones
## observaciones   0   1
##             0 395  46
##             1  67 224
```

```
mosaic(M_C, shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"), 2, 2)))
```




```
Ac = (M_C[1,1]+M_C[2,2])/sum(M_C)
cat("La Exactitud (accuracy) del modelo es", Ac,"\n")
```

```
## La Exactitud (accuracy) del modelo es 0.8456284
```

```
Se = M_C[1,1]/sum(M_C[1,])
cat("La Sensibilidad del modelo es", Se,"\n")
```

```
## La Sensibilidad del modelo es 0.8956916
```

```
Sp = M_C[2,2]/sum(M_C[2,])
cat("La Especificidad del modelo es", Sp,"\n")
```

```
## La Especificidad del modelo es 0.7697595
```

```
P = M_C[1,1]/sum(M_C[,1])
cat("La Precisión del modelo es", P,"\n")
```

```
## La Precisión del modelo es 0.8549784
```

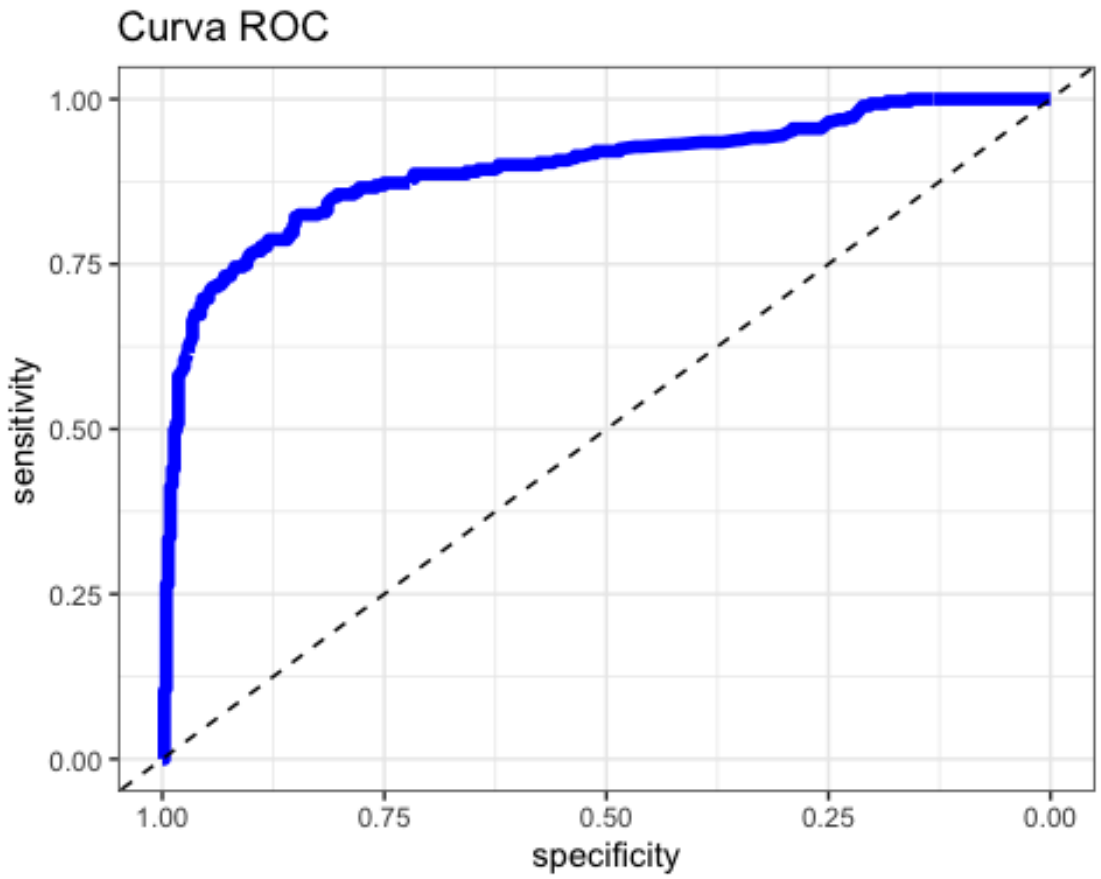
Define si el modelo es bueno o no.

El modelo es bueno por sus buenos valores de exactitud, sensibilidad y precision, donde podria mejorar es en especificidad que es clasificar mejor a los no sobrevivientes resultando en menos falsos negativos.

Curva ROC

Para hacer la curva, es necesario crear las predicciones para el data set de entrenamiento. El comando *roc* cálculará la sensibilidad y la especificidad para los datos obtenidos.

```
pred = predict(C, data = M_train, type = 'response')

library(pROC)

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

##
## Attaching package: 'pROC'

## The following object is masked from 'package:Metrics':
##
##     auc

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

ROC <- roc(response=M_train$Survived, predictor=pred)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

ROC

##
## Call:
## roc.default(response = M_train$Survived, predictor = pred)
##
## Data: pred in 441 controls (M_train$Survived 0) < 291 cases
(M_train$Survived 1).
## Area under the curve: 0.8939

ggroc(ROC, color = "blue", size = 2) + geom_abline(slope = 1, intercept = 1,
linetype ='dashed') + labs(title = "Curva ROC") + theme_bw()
```



Nota: Se grafica Especificidad, pero en realidad se está graficando 1 - Especificidad.

Interpreta el gráfico y la salida que da el comando *roc*

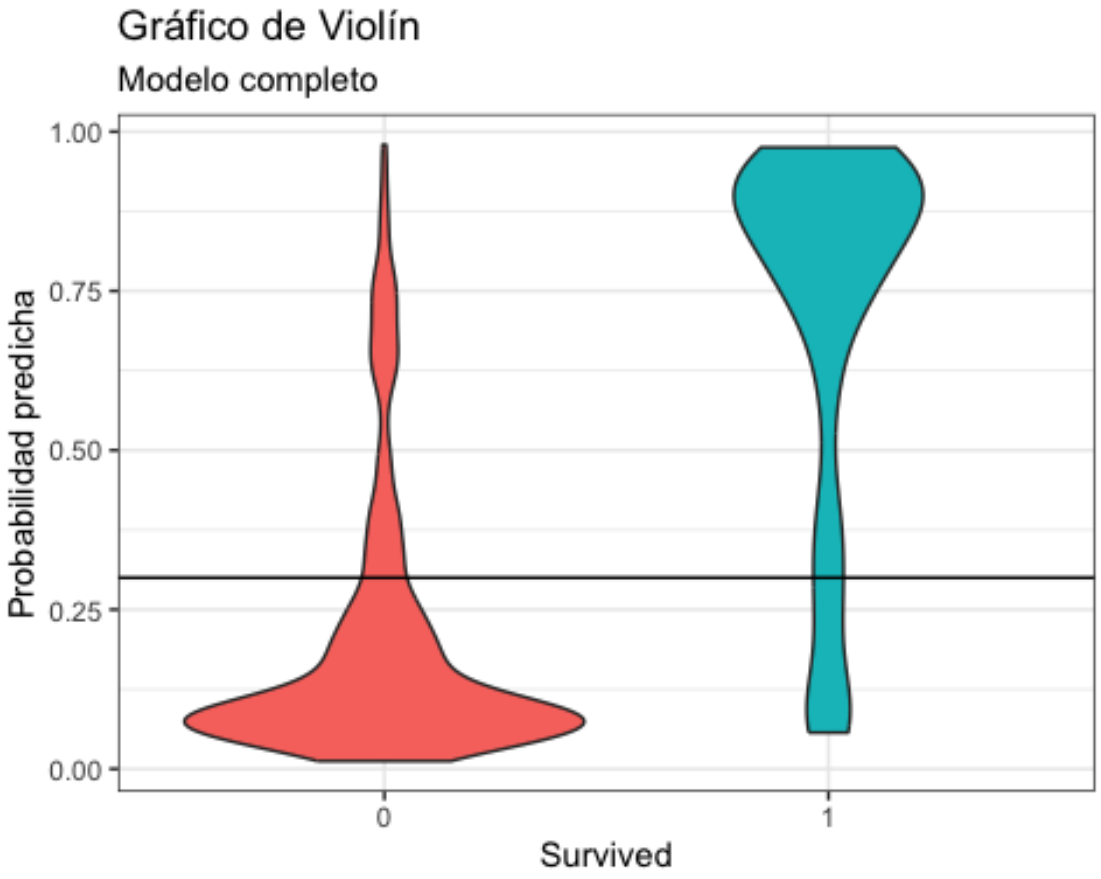
El comando da un resultado alto por lo tanto si se esta haciendo buen trabajo en predecir correctamente.

Gráfico de violín

Se crea la base de datos para el gráfico, se usan las predicciones ya elaboradas para el gráfico ROC y las clasificaciones originales (*train$M_Survived*).

```
v_d = data.frame(Survived=M_train$Survived,pred=pred)

ggplot(data=v_d, aes(x=Survived, y=pred, group=Survived,
fill=factor(Survived))) +
  geom_violin() + geom_abline(aes(intercept=0.3,slope=0))+
  theme_bw() +
  guides(fill=FALSE) +
  labs(title='Gráfico de Violín', subtitle='Modelo completo', y='Probabilidad
predicha')

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use
"none" instead as
```

```
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Interpreta

Validación

Elección de un umbral de clasificación optimo.

Elección del umbral de clasificación (punto de corte)

Se trabaja con la base de datos de validación (*M_valid*) y se realiza el gráfico de la Exactitud, Sensibilidad, Especificidad y Precisión para distintos valores del umbral de clasificación. Se siguen los siguientes pasos:

1. Predicción en los datos de validación con el modelo elegido (en el ejemplo, el B)
2. Se definen los umbrales de clasificación: irán desde 0.05 hasta 0.95.
3. Se definen las métricas de la matriz de confusión para cada umbral de clasificación

4. Se prepara el conjunto de datos: se quitan los NA y se agrega la columna de umbrales de clasificaición
5. Se le da un formato a la base de datos para que pueda ser graficada más fácilmente.

Generación de base de datos para graficar

```
pred_val = predict(C, newdata=M_valid, type='response')
clase_real = M_valid$Survived

datosV = data.frame(accuracy=NA, recall=NA, specificity = NA, precision=NA)

for (i in 5:95){
  clase_predicha = ifelse(pred_val>i/100,1,0)

##Creamos la matriz de confusión
cm= table(clase_predicha,clase_real)

## AccurAcy: Proporción de correctamente predichos
datosV[i,1] = (cm[1,1]+cm[2,2])/(cm[1,1]+cm[1,2]+cm[2,1]+cm[2,2])
## Recall: Tasa de positivos correctamente predichos
datosV[i,2] = (cm[2,2])/(cm[1,2]+cm[2,2])
## Specificity: Tasa de negativos correctamente predichos
datosV[i,3] = cm[1,1]/(cm[1,1]+cm[2,1])
## Precision: Tasa de bien clasificados entre los clasificados como positivos
datosV[i,4] = cm[2,2]/(cm[2,1]+cm[2,2])
}

## Se limpia el conjunto de datos
datosV = na.omit(datosV)
datosV$umbral = seq(0.05,0.95,0.01)
```

Formato de datos

- Se crea la variable *métrica* que será una variable categórica para las métricas (Exactitud, Sensibilidad, Especificidad y Precisión)
- Los valores de las métricas se ponen en una sola columna.

- Se identifican las métricas para los distintos umbrales con la variable 'umbral'.

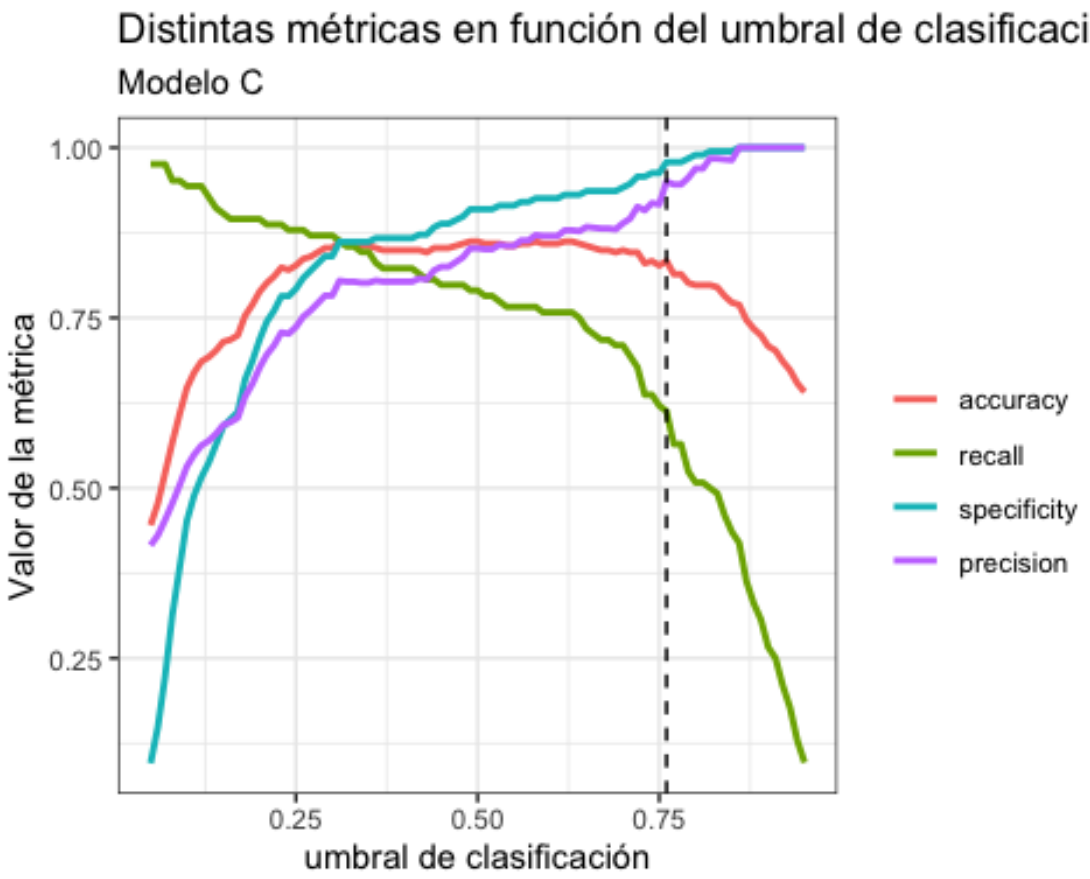
```
library(reshape2)

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

datosV_m <- reshape2::melt(datosV,id.vars=c('umbral'))
colnames(datosV_m)[2] <- c('Metrica')
```

Gráfica

En la gráfica se define cuál es el mejor umbral de clasificación dependiendo de cuál métrica es más importante en el contexto del problema (Exactitud, Sensibilidad, Especificidad o Precisión). Si no hay una métrica de preferencia, se opta por escoger el máximo valor de que pueden tener estás métricas en conjunto. En cualquier caso da valores a u para mover el umbral de clasificación y observar como se comporta con respecto a las métricas.

```
library(ggplot2)

u = 0.76

ggplot(data=datosV_m, aes(x=umbral,y=value,color=Metrica)) +
geom_line(size=1) + theme_bw() +
  labs(title= 'Distintas métricas en función del umbral de clasificación',
       subtitle= 'Modelo C',
       color="", x = 'umbral de clasificación', y = 'Valor de la métrica') +
  geom_vline(xintercept=u, linetype="dashed", color = "black")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

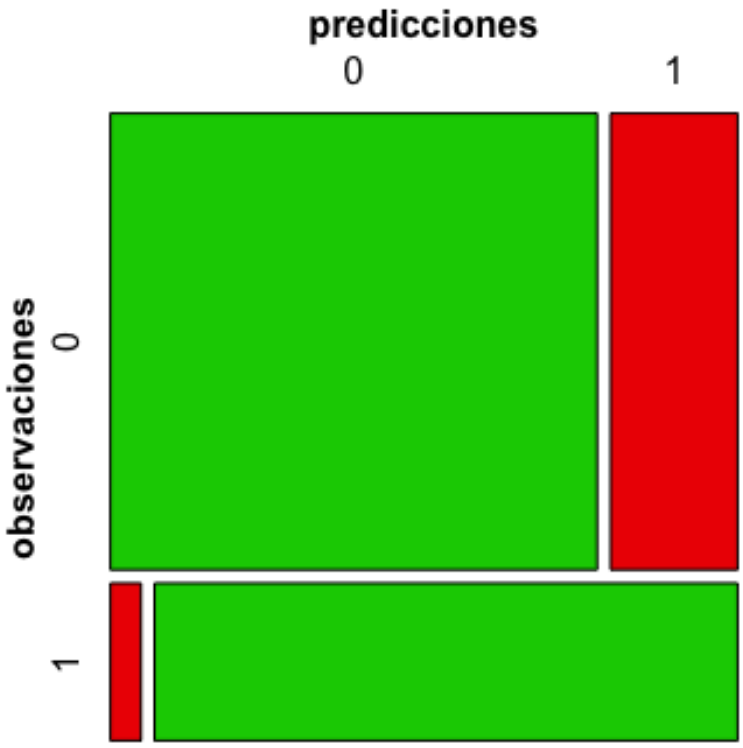


Define cuál es el mejor umbral en donde se obtienen las mejores métricas Recall, Accuracy, Sensitivity y Specificity.

Matriz de confusión con el umbral de clasificación optimo

De acuerdo al umbral seleccionado, calcula la matriz de confusión y las métricas obtenidas. Indica si mejora la predicción con respecto al umbral de u = 0.5, que es el que se maneja por default.

```
prediccionesV = ifelse(pred_val > 0.76, yes = 1, no = 0)
M_Cv <- table(prediccionesV, M_valid$Survived, dnn = c("observaciones",
"predicciones"))
M_Cv

##              predicciones
## observaciones   0   1
##             0 184  48
##             1   4  76

mosaic(M_Cv, shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"), 2, 2)))
```




```
AcV = (M_Cv[1,1]+M_Cv[2,2])/sum(M_Cv)
cat("La Exactitud (accuracy) del modelo es", AcV,"\n")

## La Exactitud (accuracy) del modelo es 0.8333333

SeV = M_Cv[1,1]/sum(M_Cv[1,])
cat("La Sensibilidad del modelo es", SeV,"\n")

## La Sensibilidad del modelo es 0.7931034

SpV = M_Cv[2,2]/sum(M_Cv[2,])
cat("La Especificidad del modelo es", SpV,"\n")

## La Especificidad del modelo es 0.95

PV = M_Cv[1,1]/sum(M_Cv[,1])
cat("La Precisión del modelo es", PV,"\n")

## La Precisión del modelo es 0.9787234
```

Testeo

Calcula la matriz de confusión con los datos de prueba y el umbral de clasificación seleccionado. Indica que tan bueno es tu modelo y con él tu umbral de clasificación seleccionado.

Conclusiones

Concluye definiendo cuáles fueron las principales características de las personas que sobrevivieron e indica cuáles son los coeficientes de cada variable en el modelo de predicción de sobrevivencia.

Interpreta los coeficientes de predicción de cada variable. Indica cómo influyó en la sobrevivencia.

```
          Df Deviance    AIC
```

- Embarked 2 558.57 574.57
- PassengerId 1 558.36 576.36
- Parch 1 558.47 576.47 557.66 577.66
- SibSp 1 563.07 581.07
- Age 1 572.12 590.12
- Pclass 2 599.58 615.58
- Sex 1 885.72 903.72

Coefficients: Estimate Std. Error z value Pr(>|z|)
(Intercept) 4.165005 0.478903 8.697 < 2e-16 ***Pclass2 -1.135401 0.311629 -3.643 0.000269*** Pclass3 -2.052190 0.300851 -6.821 9.02e-12 ***Sexmale -3.632847 0.245192 -14.816 < 2e-16*** Age -0.032774 0.008596 -3.813 0.000137 ** *SibSp -0.318018 0.142805 -2.227 0.025952*
Parch -0.112929 0.129372 -0.873 0.382716

Las Pclass2 y 3, podemos deducir que los pasajeros de tercera clase tienen menos probabilidades de supervivencia que el otro. Dentro de Sexmale podemos ver que el ser hombre afecta altamente la probabilidad de supervivencia en comparacion a la mujer. Age tiene un impacto no tan grande pero es significativo. SibSp teniendo una cantidad mayor de hermanos o esposos, disminuye la probabilidad de supervivencia. Parch no es tan significativo ya que su pvalue es mayor a 0.05.

De estas variables las 3 mas valiosas son sex, pclass y age.

Indica cuál es el mejor umbral de clasificación y por qué.

El mejor umbral para el modelo que seleccione fue de 0.76, el metodo que elegi es tener el mayor numero en conjunto de estas variables ya que no tenia en mente solo maximizar una si no encontrar un equilibrio bueno, de ese umbral los resultados fueron los siguientes, los cuales nos dan valores buenos:

La Exactitud (accuracy) del modelo es 0.8557692 La Sensibilidad del modelo es 0.826484 La Especificidad del modelo es 0.9247312 La Precisión del modelo es 0.962766