Regresión Logística. El titanic

Nombre del estudiante

2024-11-19

Bibliotecas

```
# Cargamos todas las librería en la lista "librerias"
librerias = c('tidyverse','broom','ISLR','GGally','modelr','cowplot','rlang','modelr','tibble','Metrics','mic
e','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 1
- *PassengerId:* Ids del pasajero 2
- *Survived:* Si sobrevivió o no (No = 0, Sí = 1) 3
- *Ticket:* Número de ticket 4
- *Cabin:* Cabina en la que viajó 5
- *Pclass:* Clase en la que viajó (1 = 1era, 2 = 2da, 3 = 3ra) 6
- *Sex:* Masculino o Femenino (male/female) 7
- *Age:* Edad 8
- *SibSp:* Número de hermanos/conyuge a bordo 9
- *Parch:* Número de padres/hijos a bordo 10
- *Fare:* Tarifa que pagó 11
- *Embarked:* Puerto de embarcación (C = Cherbourg, Q = Queenstown, S = Southampton) 12

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)

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,-11, -1)]

#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
```

0
0
0
NA
0
0
NA
NA

Ninguna variable contiene espacios vacíos, pero las variables 4 (Age), 7 (Fare) y 8 (Embarked) tienen datos faltantes.

Para contar los datos faltantes:

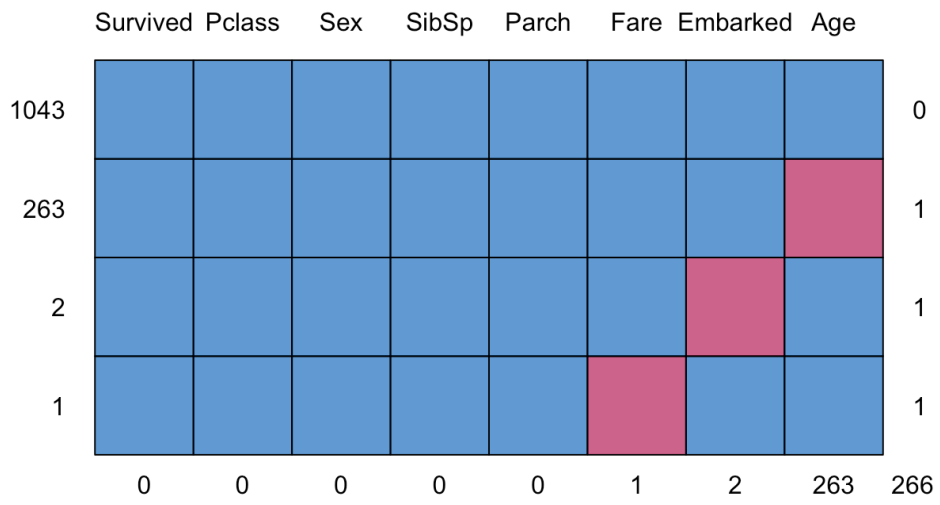
```
N = apply(X=is.na(M1),MARGIN = 2,FUN = sum)
P = round(100*N/length(M1[,2]),2)
NP = data.frame(as.numeric(N),as.numeric(P))
row.names(NP)= c( "Survived", "Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Embarked")
names(NP)=c("Número","Porcentaje")
t(NP)
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
Número	0	0	0	263.00	0	0	1.00	2.00
Porcentaje	0	0	0	20.09	0	0	0.08	0.15

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

Observemos el patrón de los datos faltantes:

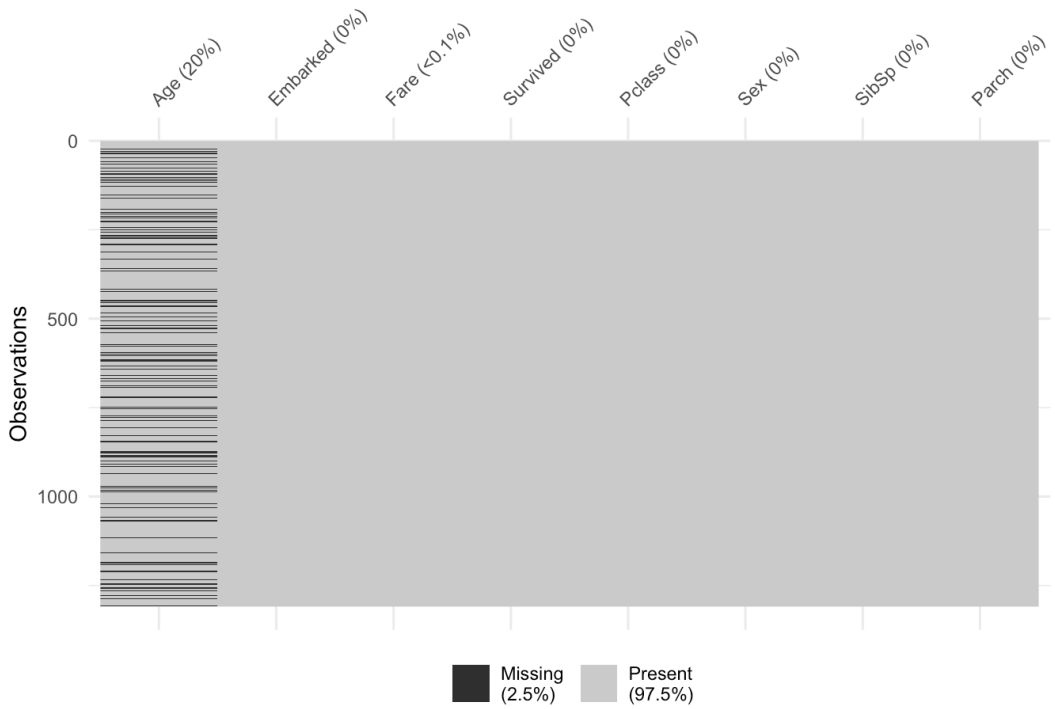
```
md.pattern(M1)
```



	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	Age	
1043	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
1	1	1	1	1	1	0	1	1	1
	0	0	0	0	0	1	2	263	266

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])
```

Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
1:323	female:466	Min. : 0.17	Min. :0.0000	Min. :0.000	Min. : 0.000	C :270
2:277	male :843	1st Qu.:21.00	1st Qu.:0.0000	1st Qu.:0.000	1st Qu.: 7.896	Q :123
3:709	NA	Median :28.00	Median :0.0000	Median :0.000	Median : 14.454	S :914
NA	NA	Mean :29.88	Mean :0.4989	Mean :0.385	Mean : 33.295	NA's: 2
NA	NA	3rd Qu.:39.00	3rd Qu.:1.0000	3rd Qu.:0.000	3rd Qu.: 31.275	NA
NA	NA	Max. :80.00	Max. :8.0000	Max. :9.000	Max. :512.329	NA
NA	NA	NA's :263	NA	NA	NA's :1	NA

Medidas sin datos faltantes

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

Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
1:282	female:386	Min. : 0.17	Min. :0.0000	Min. :0.0000	Min. : 0.00	C:212
2:261	male :657	1st Qu.:21.00	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.: 8.05	Q: 50
3:500	NA	Median :28.00	Median :0.0000	Median :0.0000	Median : 15.75	S:781
NA	NA	Mean :29.81	Mean :0.5043	Mean :0.4219	Mean : 36.60	NA
NA	NA	3rd Qu.:39.00	3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.: 35.08	NA
NA	NA	Max. :80.00	Max. :8.0000	Max. :6.0000	Max. :512.33	NA

Casi no difieren las medidas de las variables numéricas, sin embargo, para las variables categóricas, sí se ven afectadas las proporciones de los valores.

Sobrevivientes

```
t2c = 100*prop.table(table(M1[,1]))
t2s = 100*prop.table(table(M2[,1]))
t2p = c(t2s[1]/t2c[1],t2s[1]/t2c[1])
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	60.21	0.97
Sobrevivió	37.74	39.79	0.97

Esta variable casi no se ve afectada.

Clase en que viajó

```
t3c = 100*prop.table(table(M1[,2]))
t3s = 100*prop.table(table(M2[,2]))
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	27.04	1.10
Segunda	21.16	25.02	1.18
Tercera	54.16	47.94	0.89

Esta variable sí se ve un poco más afectada que la anterior, las proporciones varían más del 10%.

Sexo

```
t4c = 100*prop.table(table(M1[,3]))
t4s = 100*prop.table(table(M2[,3]))
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	37.01	1.04
Hombre	64.4	62.99	0.98

Esta variable no se ve tan afectada.

Puerto de embarcación

```
t9c = 100*prop.table(table(M1[,8]))
t9s = 100*prop.table(table(M2[,8]))
t9p = c(t9s[1]/t9c[1],t9s[2]/t9c[2],t9s[3]/t9c[3])
t9 = data.frame(as.numeric(t9c),as.numeric(t9s),as.numeric(t9p))
row.names(t9) = c("Cherbourg","Queenstown","Southampton")
names(t9) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t9,2)
```

	Con NA (%)	Sin NA (%)	Pérdida (prop)
Cherbourg	20.66	20.33	0.98
Queenstown	9.41	4.79	0.51
Southampton	69.93	74.88	1.07

Esta variable tampoco se ve tan afectada a pesar de que Queenstown pierde el 50% de los valores, ya eran muy pocos en un principio.

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()
```

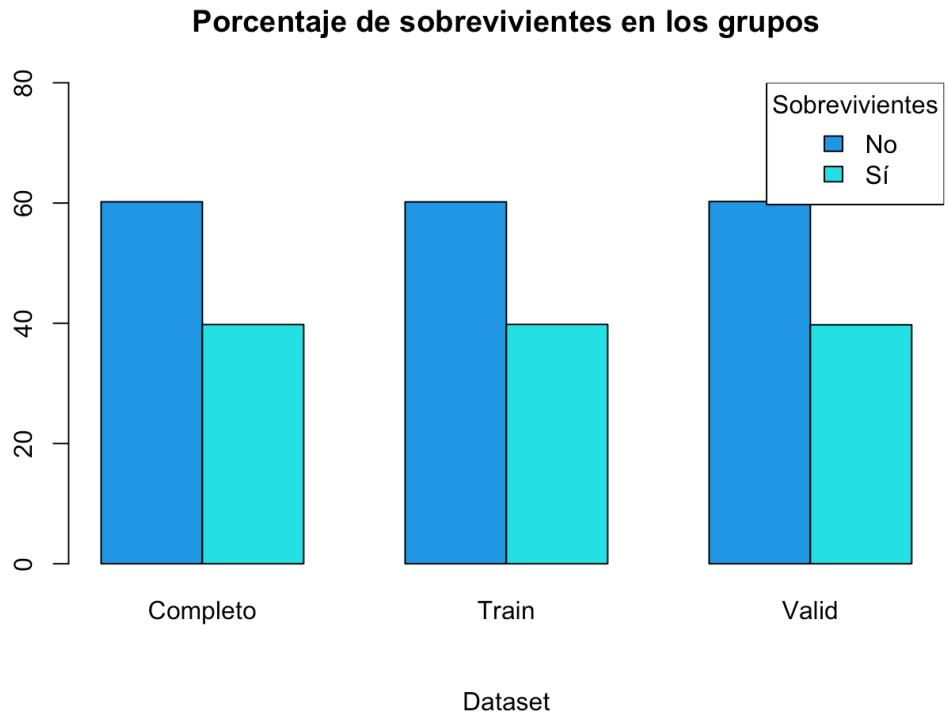
Proporciones de sobrevivientes en las tres bases de datos

- 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:

```
propCompleta = prop.table(table(M2$Survived))
propTrain = prop.table(table(M_train$Survived))
propValid = prop.table(table(M_valid$Survived))

TablaComparativa <- data.frame(
  Survived = names(propCompleta),
  ProporcionCompleta = as.numeric(propCompleta),
  ProporcionTrain = as.numeric(propTrain),
  ProporcionTest = as.numeric(propValid)
)

print(TablaComparativa)
```

```
##   Survived ProporcionCompleta ProporcionTrain ProporcionTest
## 1        0          0.6021093       0.6019152      0.6025641
## 2        1          0.3978907       0.3980848      0.3974359
```

```
TablaComparativaMatrix <- as.matrix(TablaComparativa[, -1])

# Bar plot
barplot(
  TablaComparativaMatrix * 100,
  col = 4:5,
  beside = TRUE,
  main = "Porcentaje de sobrevivientes en los grupos",
  sub = "Dataset",
  ylim = c(0, 80),
  names.arg = c("Completo", "Train", "Valid")
)

# Add a legend
legend(
  "topright",
  legend = c("No", "Sí"),
  title = "Sobrevivientes",
  fill = 4:5
)
```



La proporción de sobrevivientes se mantiene a lo largo de los 3 datasets.

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 ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked, data = M_train, family = "binomial")
```

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

```
## Start:  AIC=533.65
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked
##
##            Df Deviance    AIC
## - Embarked  2   513.66 529.66
## - Parch     1   514.43 532.43
## - Fare      1   515.06 533.06
## <none>          513.65 533.65
## - SibSp     1   519.77 537.77
## - Age       1   534.08 552.08
## - Pclass    2   541.38 557.38
## - Sex       1   866.04 884.04
##
## Step:  AIC=529.66
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare
##
##            Df Deviance    AIC
## - Parch     1   514.45 528.45
## - Fare      1   515.08 529.08
## <none>          513.66 529.66
## + Embarked  2   513.65 533.65
## - SibSp     1   519.82 533.82
## - Age       1   534.17 548.17
## - Pclass    2   542.31 554.31
## - Sex       1   871.49 885.49
##
## Step:  AIC=528.45
## Survived ~ Pclass + Sex + Age + SibSp + Fare
##
##            Df Deviance    AIC
## - Fare      1   515.45 527.45
## <none>          514.45 528.45
## + Parch     1   513.66 529.66
## + Embarked  2   514.43 532.43
## - SibSp     1   522.07 534.07
## - Age       1   534.29 546.29
## - Pclass    2   544.68 554.68
## - Sex       1   878.35 890.35
##
## Step:  AIC=527.45
## Survived ~ Pclass + Sex + Age + SibSp
##
##            Df Deviance    AIC
## <none>          515.45 527.45
## + Fare      1   514.45 528.45
## + Parch     1   515.08 529.08
## + Embarked  2   515.42 531.42
## - SibSp     1   522.43 532.43
## - Age       1   535.91 545.91
## - Pclass    2   569.01 577.01
## - Sex       1   891.03 901.03
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = "binomial",
##     data = M_train)
##
## Coefficients:
## (Intercept)      Pclass2      Pclass3      Sexmale          Age        SibSp
##     4.62056     -1.29166     -2.20695     -3.89311     -0.04052     -0.34474
##
## Degrees of Freedom: 730 Total (i.e. Null);  725 Residual
## Null Deviance:       982.8
## Residual Deviance: 515.5     AIC: 527.5
```

- 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.

El mejor modelo es tomando en cuenta las variables SibSp, Age, Pclass y Sex, obtuvo un valor para el AIC de 550.1. La ultima variable que eliminó fue Parch.

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)

Este modelo incluye las variables Pclass + Sex + Age + SibSp + Parch.

Para evaluar el modelo se deben tomar en cuenta las hipótesis:

Hipótesis de variables

$H_0 : \beta_i = 0$

$H_1 : \beta_i \neq 0$

Hipótesis de modelo

$H_0 : \beta_1 = \beta_2 = \cdots = \beta_n = 0$

$H_1 : \text{Al menos un } \beta_i \neq 0$

```
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)  4.702642   0.529891   8.875  < 2e-16 ***
## Pclass2     -1.303015   0.329400  -3.956 7.63e-05 ***
## Pclass3     -2.216407   0.317867  -6.973 3.11e-12 ***
## Sexmale     -3.928027   0.261137 -15.042  < 2e-16 ***
## Age         -0.041334   0.009346  -4.423 9.75e-06 ***
## SibSp       -0.323188   0.138458  -2.334   0.0196 *
## Parch       -0.087253   0.143108  -0.610   0.5421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 982.80  on 730  degrees of freedom
## Residual deviance: 515.08  on 724  degrees of freedom
## AIC: 529.08
##
## Number of Fisher Scoring iterations: 5
```

De acuerdo con las pruebas de hipótesis, el modelo es válido pero no todas las variables son significativas, la variable Parch obtiene un valor p de 0.21, debido a que es mayor al alpha definido, esta variable no se considera significativa.

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)

Las variables de este modelo son: Pclass + Sex + Age + SibSp

Para evaluar las variables y el modelo se consideran las mismas pruebas de hipótesis que en el punto anterior.

```
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)  4.620560   0.509686   9.065  < 2e-16 ***
## Pclass2     -1.291657   0.328752  -3.929 8.53e-05 ***
## Pclass3     -2.206947   0.317548  -6.950 3.65e-12 ***
## Sexmale     -3.893115   0.253236 -15.373  < 2e-16 ***
## Age         -0.040521   0.009223  -4.393 1.12e-05 ***
## SibSp       -0.344739   0.134064  -2.571   0.0101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 982.80  on 730  degrees of freedom
## Residual deviance: 515.45  on 725  degrees of freedom
## AIC: 527.45
##
## Number of Fisher Scoring iterations: 5
```

En este caso el modelo y todas las variables son significativas. Se obtiene un AIC de 540.78, una desviación nula de 982.80, desviación residual de 528.78. La diferencia entre los valores de las desviaciones parece indicar que el modelo funciona correctamente, de lo contrario, los valores serían similares.

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:

```
aic_B <- B$aic
deviance_B <- B$deviance
null_deviance_B <- B$null.deviance

aic_C <- C$aic
deviance_C <- C$deviance
null_deviance_C <- C$null.deviance

TablaComparativa <- data.frame(
  Metric = c("AIC", "Residual Deviance", "Null Deviance"),
  Model_B = c(aic_B, deviance_B, null_deviance_B),
  Model_C = c(aic_C, deviance_C, null_deviance_C)
)

print(TablaComparativa)
```

```
##              Metric  Model_B  Model_C
## 1               AIC 529.0801 527.4510
## 2 Residual Deviance 515.0801 515.4510
## 3     Null Deviance 982.7966 982.7966
```

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

La desviación nula en ambos modelos es la misma porque es la predicción sin utilizar ninguna de las variables independientes, por lo que tiene sentido que sean iguales ya que no se están considerando estas variables.

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

El AIC es un tanto menor en el modelo C, considerando que en este criterio un menor valor es mejor, el modelo C es superior al B. La desviación residual es menor en el modelo B, esto puede ser debido a que tiene más información ya que cuenta con una variable más, sin embargo, esta diferencia es marginal y puede no ser tan relevante.

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

```
# Calcular pseudo R^2 para el modelo B
pseudo_R2_B <- 1 - (B$deviance / B$null.deviance)

# Calcular pseudo R^2 para el modelo C
pseudo_R2_C <- 1 - (C$deviance / C$null.deviance)

# Crear tabla comparativa
TablaDesviacion <- data.frame(
  Modelo = c("Modelo B", "Modelo C"),
  Pseudo_R2 = c(pseudo_R2_B, pseudo_R2_C)
)

# Imprimir la tabla comparativa
print(TablaDesviacion)
```

```
##     Modelo Pseudo_R2
## 1 Modelo B 0.4759037
## 2 Modelo C 0.4755262
```

Ambos modelos obtienen una pseudo R2 similar, logran explicar un 45% de la varianza en la variable de interés.

Prueba de razón de verosimilitud

H_0 : El modelo nulo explica mejor la variable respuesta: $log(\frac{p}{1-p})$ (la probabilidad es constante)

H_1 : El modelo con predictores explica mejor la variable respuesta: $log(\frac{p}{1-p})$ que el modelo nulo

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

```
# Calculate the difference in deviance
Diferencia = C$null.deviance - C$deviance

# Calculate the degrees of freedom
gl = C$df.null - C$df.residual

# Calculate the p-value
p = pchisq(Diferencia, gl, lower.tail = FALSE)

# Output the p-value
cat('El valor p es: ', p)
```

```
## El valor p es:  8.8974e-99
```

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.

```
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
```

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

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
724	515.0801	NA	NA	NA
725	515.4510	-1	-0.3709589	0.542482

Esta prueba demuestra que no hay una diferencia significativa entre los modelos, se seleccionará el modelo C puesto que obtiene un menor AIC y además utiliza menos variables por lo que es más económico.

Modelo Seleccionado

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

```
C$coefficients
```

```
## (Intercept)     Pclass2     Pclass3     Sexmale         Age       SibSp
##  4.62055990 -1.29165706 -2.20694667 -3.89311457 -0.04052089 -0.34473927
```

```
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:

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

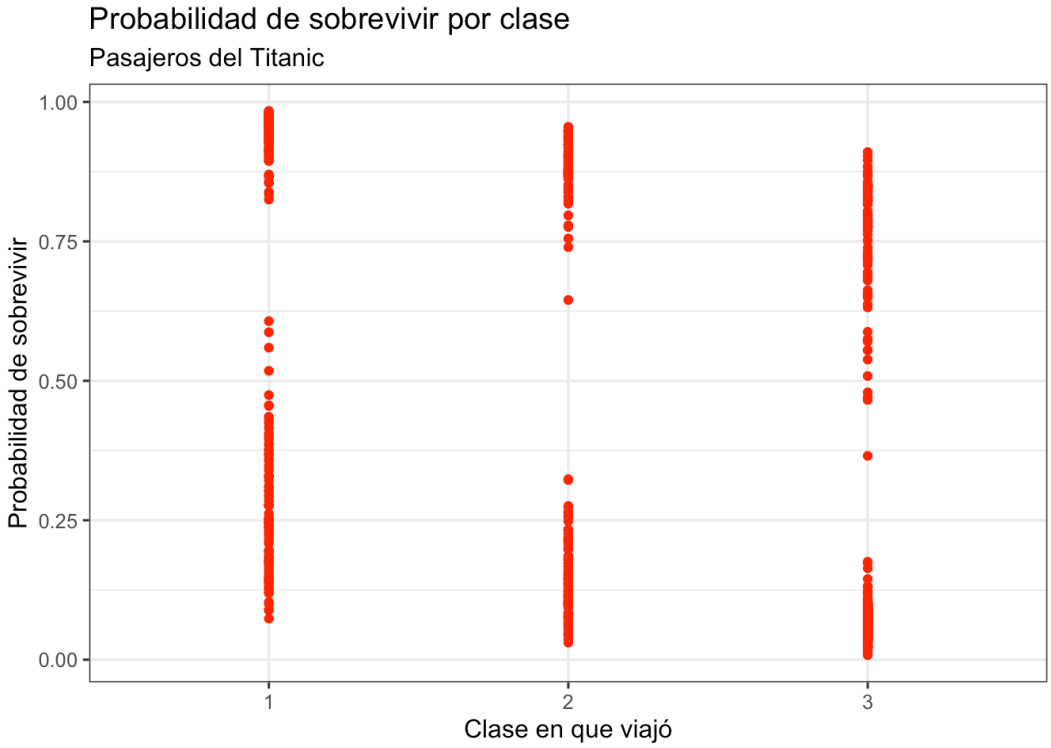
Así que interpretar el efecto de cada variable, se grafica cada una de ellas contra los valores predichos para la probabilidad de sobrevivencia.

Para hacer los gráficos se ejemplifica con:

Clase en que viajó el pasajero

```
p_pred = B$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)
```

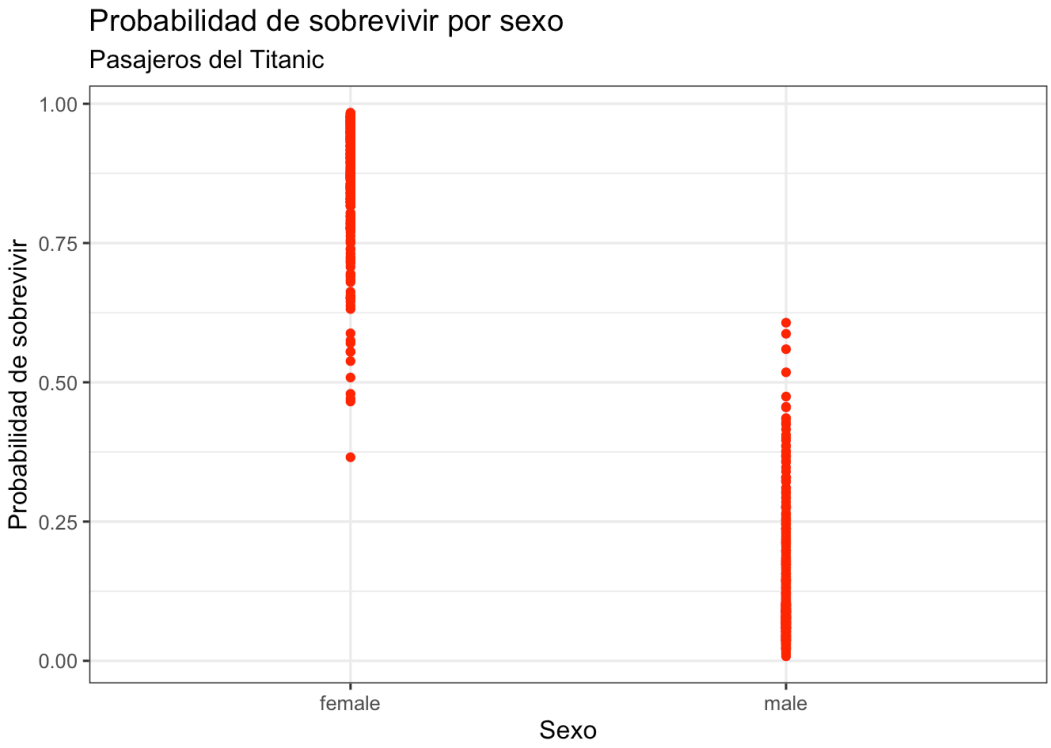
```
## Warning: Use of `M_pred$p_pred` is discouraged.
## ℹ Use `p_pred` instead.
```



Algunos pasajeros de primera clase parecen tener una probabilidad más alta de sobrevivir, pero hay más pasajeros de tercera clase que estan por encima de 0.5.

```
ggplot(M_pred, aes( x = Sex)) +
geom_point(aes(y=M_pred$p_pred), size=1.5,color="red") +
  labs(x="Sexo", y="Probabilidad de sobrevivir",
       title="Probabilidad de sobrevivir por sexo",
       subtitle="Pasajeros del Titanic",
       col="")+
  theme_bw(base_size = 12)
```

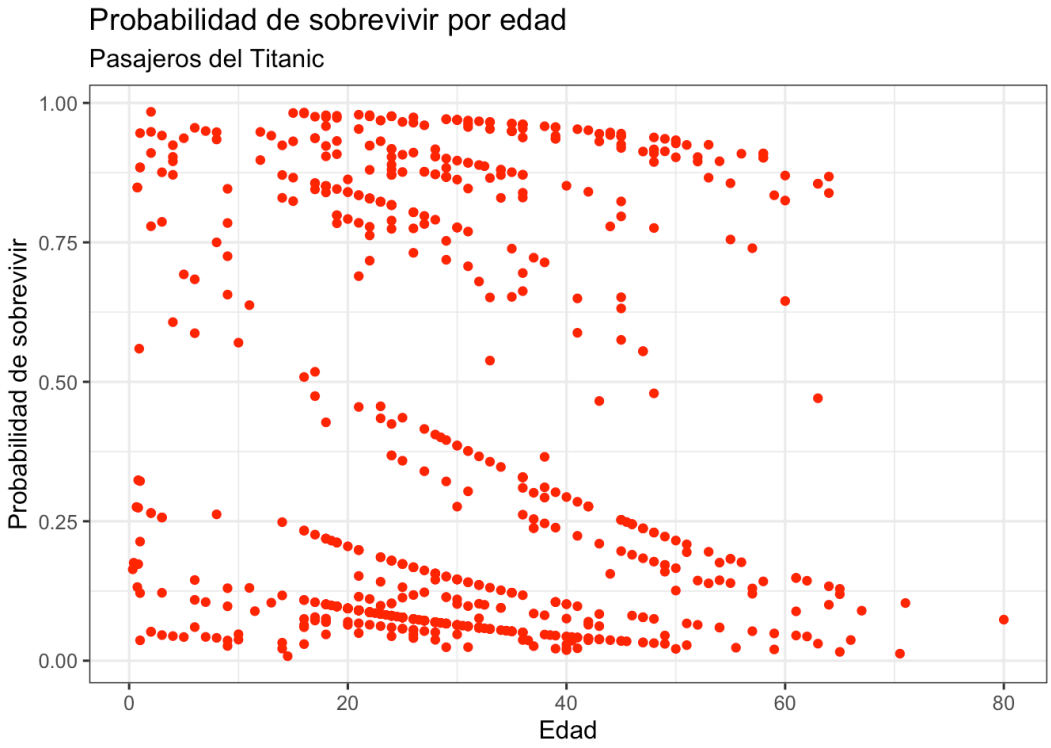
```
## Warning: Use of `M_pred$p_pred` is discouraged.
## ℹ Use `p_pred` instead.
```



En esta variable la diferencia es muy clara, las mujeres tienen más probabilidad de sobrevivir que los hombres, la mayoría de los valores para mujeres está por encima de 0.5.

```
ggplot(M_pred, aes( x = Age)) +
geom_point(aes(y=M_pred$p_pred), size=1.5,color="red") +
  labs(x="Edad", y="Probabilidad de sobrevivir",
       title="Probabilidad de sobrevivir por edad",
       subtitle="Pasajeros del Titanic",
       col="")+
  theme_bw(base_size = 12)
```

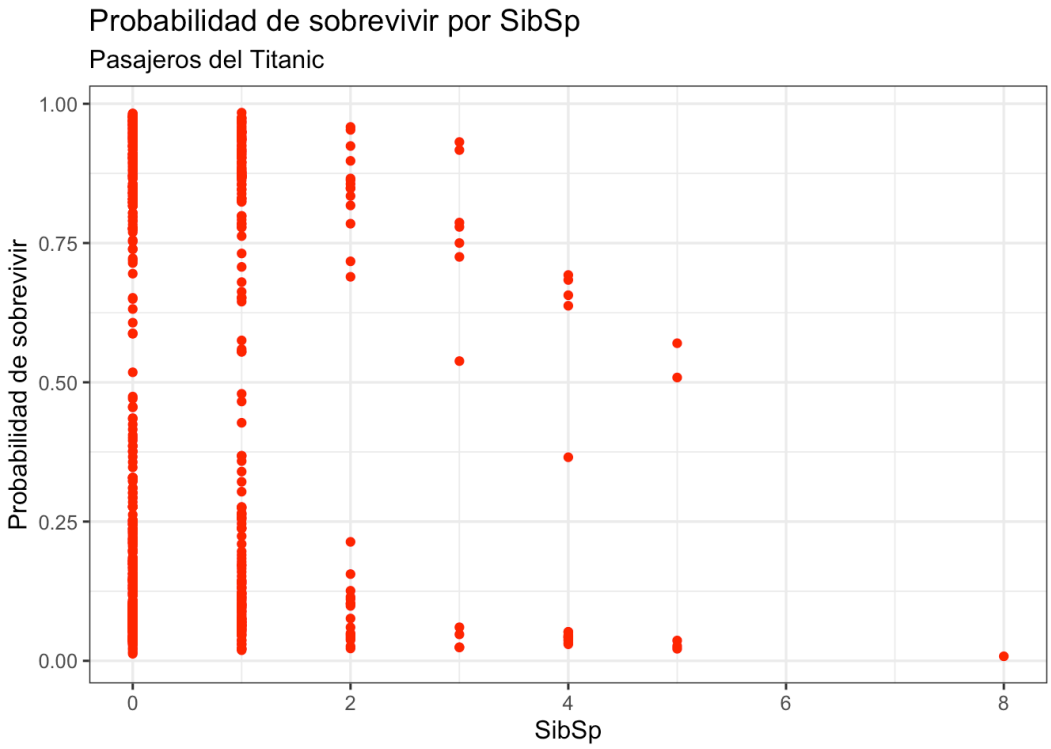
```
## Warning: Use of `M_pred$p_pred` is discouraged.
## ℹ Use `p_pred` instead.
```



Se puede observar que la probabilidad de sobrevivir conforme avanza la edad disminuye.

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

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



El efecto de esta variable es un poco menos claro, cada categoría tiene valores a lo largo de todo el rango.

Predicciones

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

Matriz de confusión

```
library(vcd)
```

```
## Loading required package: grid
```

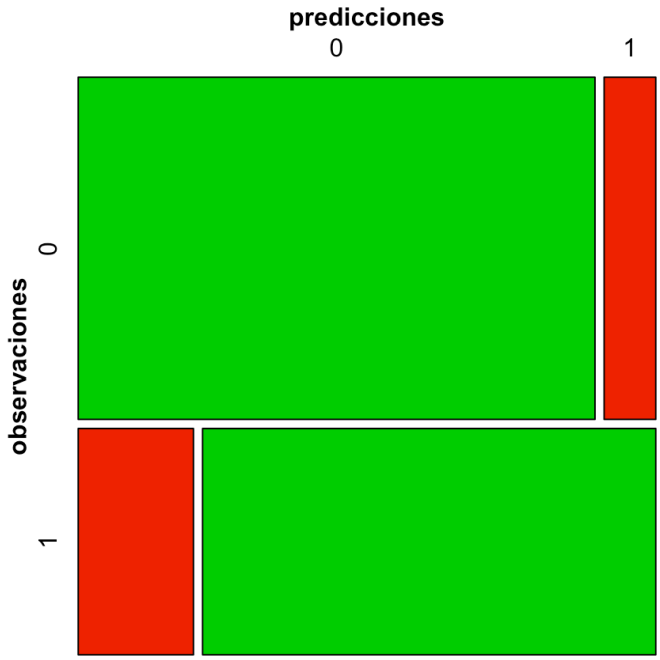
```
##
## Attaching package: 'vcd'
```

```
## 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
```

observaciones/predicciones	0	1
0	400	40
1	59	232

```
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 0.8645691
```

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

```
## La Sensibilidad del modelo es 0.9090909
```

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

```
## La Especificidad del modelo es 0.7972509
```

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

```
## La Precisión del modelo es 0.8714597
```

Define si el modelo es bueno o no.

El modelo funciona correctamente, las métricas lo confirman. Se obtuvo un 0.85 de accuracy, esto quiere decir que el 85% de los valores predichos fueron correctos. Además, el resto de las métricas como sensibilidad, especificidad y precisión también están en valores aceptables.

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)
```

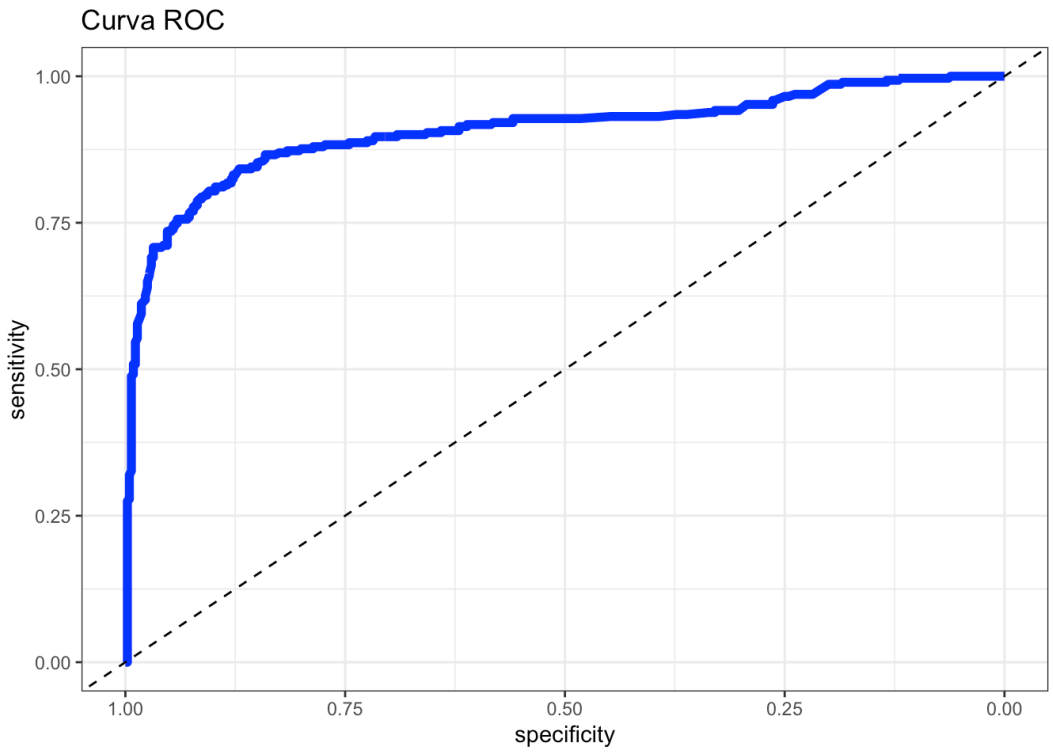
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
ROC
```

```
##
## Call:
## roc.default(response = M_train$Survived, predictor = pred)
##
## Data: pred in 440 controls (M_train$Survived 0) < 291 cases (M_train$Survived 1).
## Area under the curve: 0.9061
```

```
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*

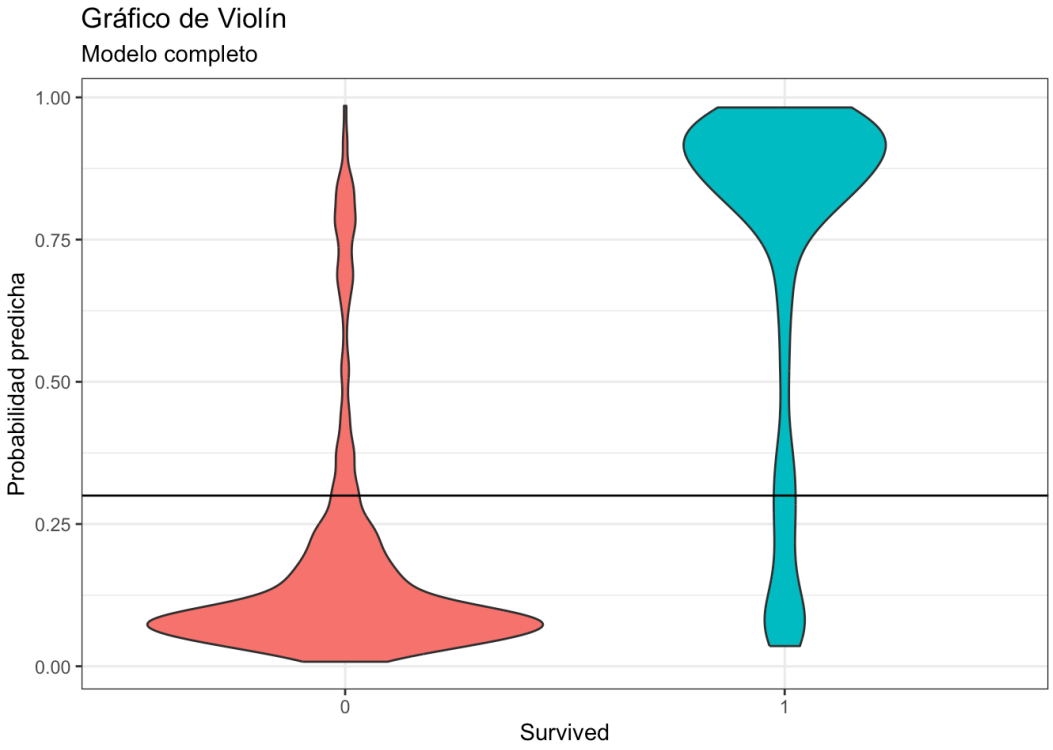
La curva ROC esta muy por encima de la diagonal, esto demuestra que el modelo funciona mejor que una predicción aleatoria, tiene un buen balance entre especificidad y sensibilidad.

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.
```



Este gráfico también demuestra que el modelo funciona, puesto que para la clase 0, la distribución tiene más valores cercanos a 0, mientras que para la clase 1 hay más valores cercanos a 1.

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
```

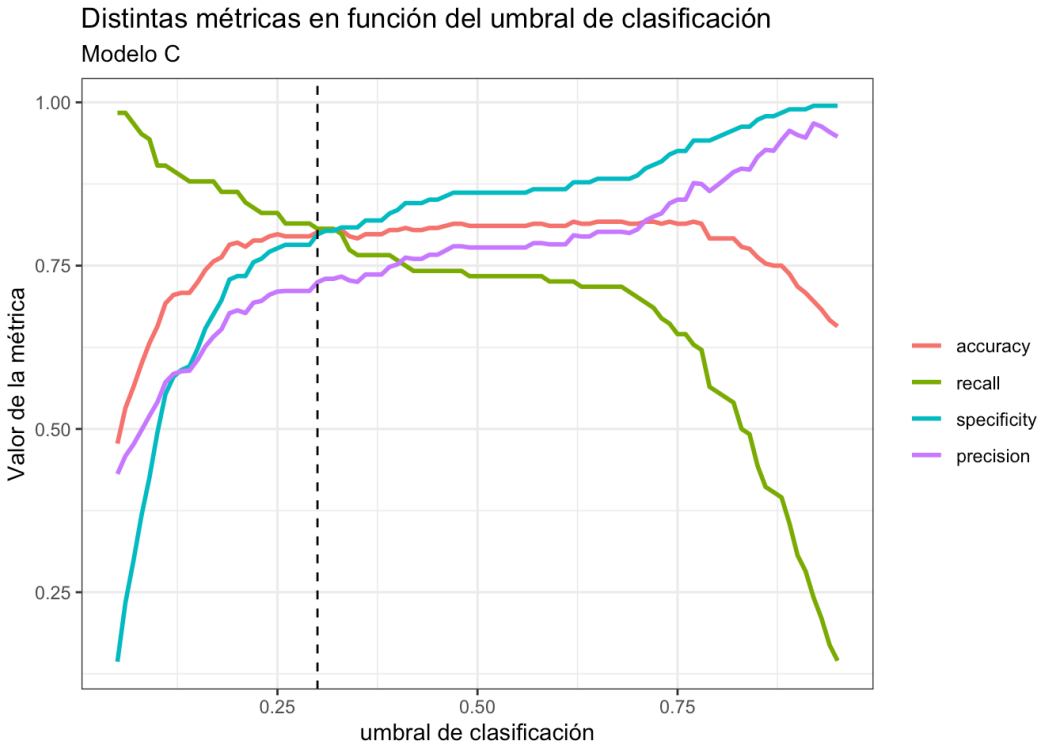
```
datosV_m <- reshape2::melt(datosV,id.vars=c('umbral'))
colnames(datosV_m)[2] <- c('Metrica')
```

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 = 0.3 #Se dio un valor arbitrario, tú modificalo de acuerdo al criterio que selecciones.

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.

El umbral seleccionado es de 0.3, que es donde se alcanza un balance entre todas las métricas

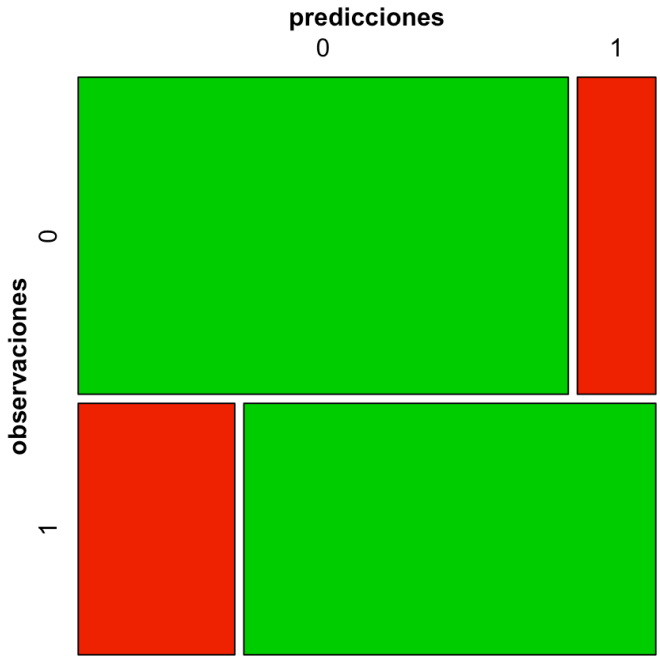
Matriz de confusión con el umbral de clasificación optimo

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 > 0.3, yes = 1, no = 0)
M_Cv <- table(prediccionesV, M_valid$Survived, dnn = c("observaciones", "predicciones"))
M_Cv
```

observaciones/predicciones	0	1
0	150	24
1	38	100

```
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 0.8012821
```

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

```
## La Sensibilidad del modelo es 0.862069
```

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

```
## La Especificidad del modelo es 0.7246377
```

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

```
## La Precisión del modelo es 0.7978723
```

Para el dataset de validación, las métricas de evaluación también son aceptables, se mantienen métricas balanceadas.

Testeo

Debido a que el dataset de testeo no cuenta con la columna de survived, solo se realiza la predicción utilizando el modelo seleccionado.

```
M_test=read.csv("Titanic_test.csv")

for(var in c('Pclass','Embarked','Sex'))
  M_test[,var] <-as.factor(M_test[,var])

M_test$Predicted_Probabilities <- predict(C, newdata = M_test, type = "response")

threshold <- 0.37
M_test$Survived <- ifelse(M_test$Predicted_Probabilities >= threshold, 1, 0)

head(M_test)
```

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Predicted_Probabilities	Survived
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292		Q	0.0532785	0
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000		S	0.5410057	1
894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875		Q	0.0440868	0
895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625		S	0.0708592	0
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875		S	0.7644826	1
897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250		S	0.1143768	0

Conclusiones

La ecuación obtenida por el modelo es la siguiente:

$$\text{logit}(P) = 4.4837 - 1.3729 \cdot \text{Pclass2} - 2.2949 \cdot \text{Pclass3} - 3.8109 \cdot \text{Sexmale} - 0.0380 \cdot \text{Age} - 0.2825 \cdot \text{SibSp}$$

Cada variable tiene un impacto distinto:

1. **Clase del pasajero (Pclass)**:
   - Los pasajeros en la **Primera Clase (Pclass1)** tenían más probabilidad de sobrevivir en comparación con los de Segunda y Tercera Clase.
   - Los coeficientes negativos de **Pclass2** (-1.3729) y **Pclass3** (-2.2949) indican que pertenecer a estas clases disminuyó la probabilidad de sobrevivencia en relación con la Primera Clase.
2. **Sexo**:
   - Ser hombre tuvo un fuerte efecto negativo en la probabilidad de sobrevivir, con un coeficiente de -3.8109. Esto indica que las mujeres tenían mucha mayor probabilidad de sobrevivir.
3. **Edad**:
   - La edad tuvo un coeficiente de -0.0380, lo que indica que a mayor edad, la probabilidad de sobrevivencia disminuye un poco.
4. **Número de hermanos o cónyuges a bordo (SibSp)**:
   - El coeficiente de -0.2825 muestra que tener más hermanos o cónyugues a bordo también disminuye ligeramente la probabilidad de sobrevivir.

El threshold que se seleccionó para realizar las predicciones fue de 0.3, puesto que en este punto se muestra un balance entre las métricas de Recall, Accuracy, Sensitivity y Specificity. Por lo que esta selección es robusta frente a falsos positivos y falsos negativos, mover el umbral hacia arriba o abajo afectaría estas medidas, dando preferencia a alguna métrica. Esto puede ser útil en otros contextos, pero en este en particular es mejor balancearlos.