Regresión Logística. El titanic

Oskar Arturo Gamboa Reyes

2024-11-19

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)

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)]

#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=9)
for(i in c(1:9)){
  V[i,] <- sum(with(M1,M1[,i])=="")}
V
```

```
  0
  0
  0
  0
 NA
  0
  0
 NA
 NA
```

Ninguna variable contiene espacios vacíos, pero las variables 5 (Age), 8 (Fare) y 9 (Embarked) tienen datos faltantes.

Para contar los datos faltantes:

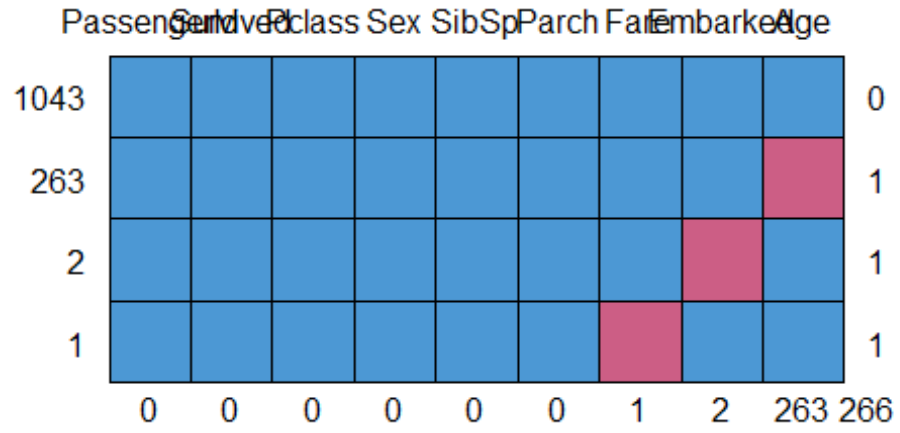
```
N = apply(X=is.na(M1),MARGIN = 2,FUN = sum)
P = round(100*N/length(M1[,2]),2)
NP = data.frame(as.numeric(N),as.numeric(P))
row.names(NP)= c("PassengerId", "Survived", "Pclass", "Sex", "Age", "SibSp",
"Parch", "Fare", "Embarked")
names(NP)=c("Número","Porcentaje")
t(NP)
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
Número	0	0	0	0	263.00	0	0	1.00	2.00
Porcentaje	0	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:

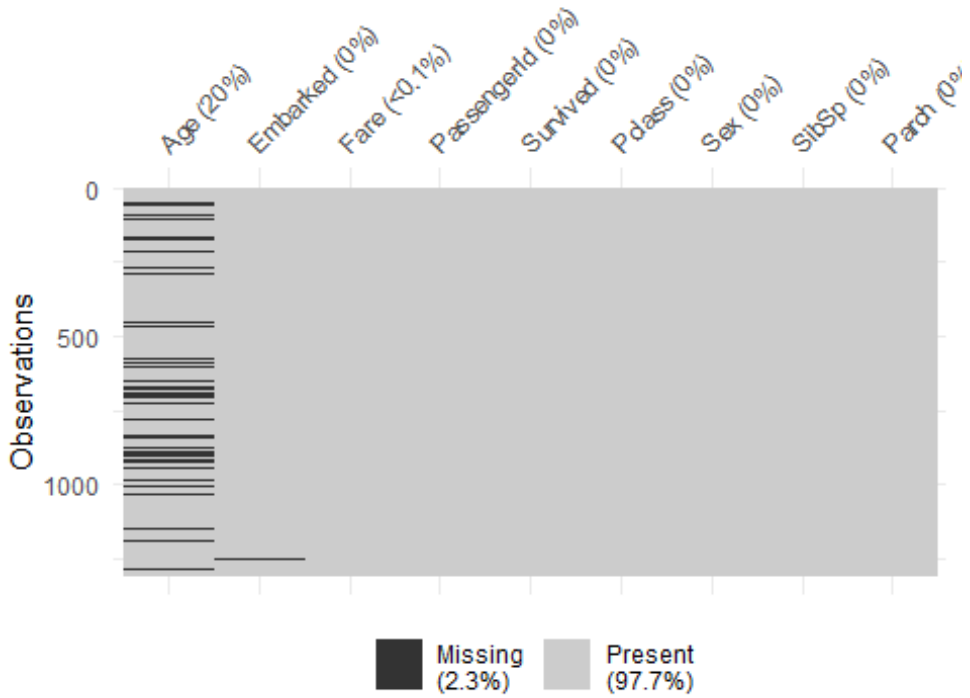
```
md.pattern(M1)
```



	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	Age	
1043	1	1	1	1	1	1	1	1	1	0
263	1	1	1	1	1	1	1	1	0	1
2	1	1	1	1	1	1	1	0	1	1
1	1	1	1	1	1	1	0	1	1	1
	0	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])
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0:815	1:323	female:466	Min. : 0.17	Min. :0.0000	Min. :0.000	Min. : 0.000	C :270
1:494	2:277	male :843	1st Qu.:21.00	1st Qu.:0.0000	1st Qu.:0.000	1st Qu.: 7.896	Q :123
NA	3:709	NA	Median :28.00	Median :0.0000	Median :0.000	Median : 14.454	S :914
NA	NA	NA	Mean :29.88	Mean :0.4989	Mean :0.385	Mean : 33.295	NA's: 2
NA	NA	NA	3rd Qu.:39.00	3rd Qu.:1.0000	3rd Qu.:0.000	3rd Qu.: 31.275	NA
NA	NA	NA	Max. :80.00	Max. :8.0000	Max. :9.000	Max. :512.329	NA
NA	NA	NA	NA's :263	NA	NA	NA's :1	NA

Medidas sin datos faltantes

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

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0:628	1:282	female:386	Min. : 0.17	Min. :0.0000	Min. :0.0000	Min. : 0.00	C:212
1:415	2:261	male :657	1st Qu.:21.00	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.: 8.05	Q: 50
NA	3:500	NA	Median :28.00	Median :0.0000	Median :0.0000	Median : 15.75	S:781
NA	NA	NA	Mean :29.81	Mean :0.5043	Mean :0.4219	Mean : 36.60	NA
NA	NA	NA	3rd Qu.:39.00	3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.: 35.08	NA
NA	NA	NA	Max. :80.00	Max. :8.0000	Max. :6.0000	Max. :512.33	NA

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	60.21	0.97
Sobrevivió	37.74	39.79	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	27.04	1.10
Segunda	21.16	25.02	1.18
Tercera	54.16	47.94	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	37.01	1.04
Hombre	64.4	62.99	0.98

Puerto de embarcación

```
t9c = 100*prop.table(table(M1[,9]))
t9s = 100*prop.table(table(M2[,9]))
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

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

Podemos ver que las variables más afectadas son Pclass y la embarcación. La información que se tiene de la clase baja se disminuye consideradamente al borrar datos faltantes, esto es probablemente porque no tenian un buen registro de estas personas. La embarcación sufre una perdida de la mitad de sus datos en Queenstown.

Análisis descriptivo

- Medidas

```
survived <- M2[M2$Survived == 1, ]
not_survived <- M2[M2$Survived == 0, ]


summary(survived)
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
	Min. : 2.0	0: 0	1:168	female :322	Min. : 0.17	Min. :0.0000	Min. :0.0000	Min. : 0.00	C:116
	1st Qu.: 340.0	1:415	2:112	male : 93	1st Qu.:19.00	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.: 13.00	Q: 20
	Median : 636.0	NA	3:135	NA	Median :28.00	Median :0.0000	Median :0.0000	Median : 26.00	S:279
	Mean : 653.3	NA	NA	NA	Mean :28.83	Mean :0.5181	Mean :0.5422	Mean : 52.95	NA
	3rd Qu.: 961.5	NA	NA	NA	3rd Qu.:37.00	3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.: 61.58	NA
	Max. :1306.0	NA	NA	NA	Max. :80.00	Max. :5.0000	Max. :5.0000	Max. :512.33	NA

```
summary(not_survived)
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
	Min. : 1.0	0:628	1:114	female : 64	Min. : 0.33	Min. :0.0000	Min. :0.0000	Min. : 0.000	C: 96
	1st Qu.:	1: 0	2:1	male	1st	1st	1st	1st Qu.:	Q:

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
313.8		49	:564	Qu.:21.00	Qu.:0.0000	Qu.:0.0000	7.896	30
Median : 674.5	NA	3:365	NA	Median :28.00	Median :0.0000	Median :0.0000	Median : 13.000	S:502
Mean : 656.7	NA	NA	NA	Mean :30.46	Mean :0.4952	Mean :0.3424	Mean : 25.799	NA
3rd Qu.: 989.5	NA	NA	NA	3rd Qu.:39.00	3rd Qu.:1.0000	3rd Qu.:0.0000	3rd Qu.: 27.721	NA
Max. :1307.0	NA	NA	NA	Max. :74.00	Max. :8.0000	Max. :6.0000	Max. :263.000	NA

```
survived_stats <- survived %>%
  summarise(
    Avg_Age = mean(Age),
    SD_Age = sd(Age),
    Avg_Fare = mean(Fare),
    SD_Fare = sd(Fare)
  )

not_survived_stats <- not_survived %>%
  summarise(
    Avg_Age = mean(Age),
    SD_Age = sd(Age),
    Avg_Fare = mean(Fare),
    SD_Fare = sd(Fare)
  )

survived_stats
```

Avg_Age	SD_Age	Avg_Fare	SD_Fare
28.82954	15.04761	52.95237	72.98792

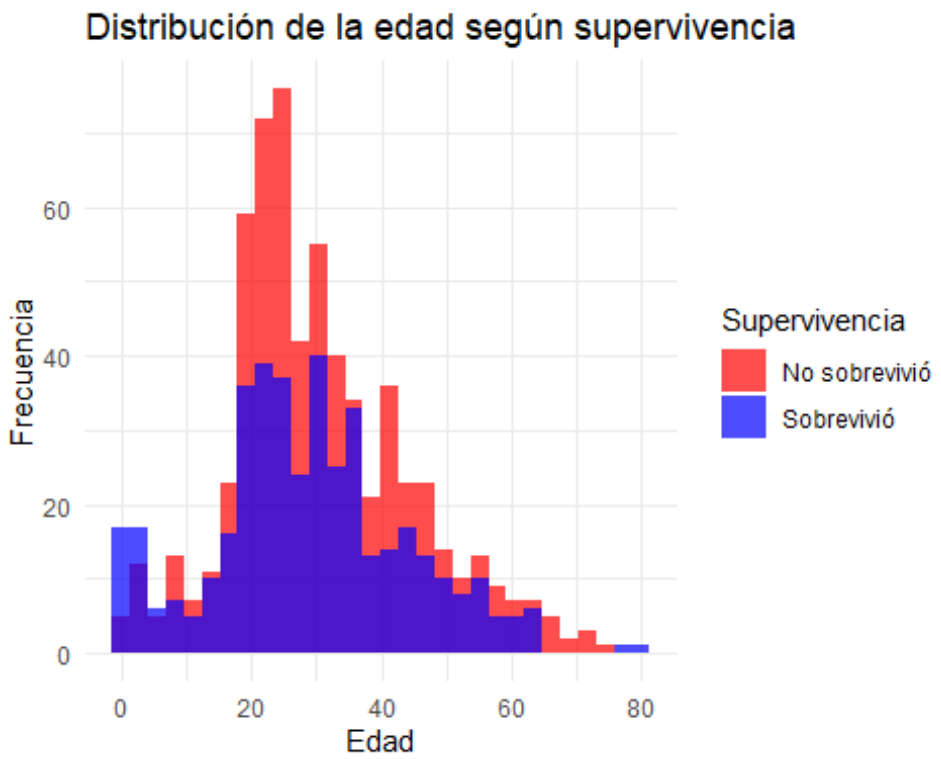
```
not_survived_stats
```

Avg_Age	SD_Age	Avg_Fare	SD_Fare
30.46323	13.87162	25.79892	36.80444

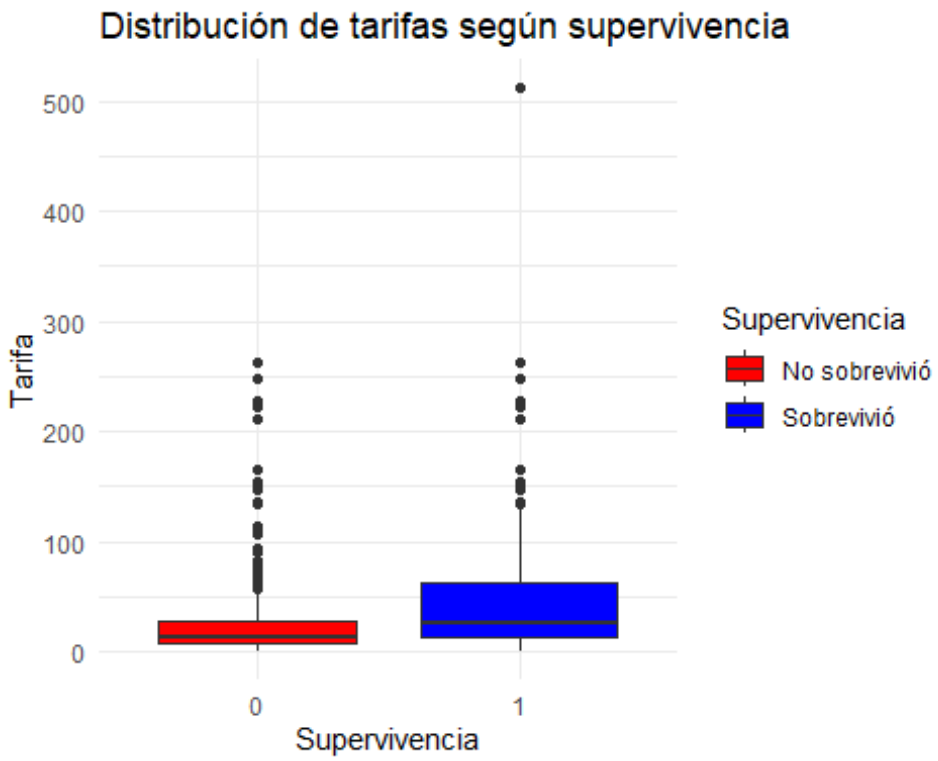
- Gráficos

```
library(ggplot2)

ggplot(M2, aes(x = Age, fill = factor(Survived))) +
  geom_histogram(bins = 30, alpha = 0.7, position = "identity") +
  scale_fill_manual(values = c("red", "blue"), labels = c("No sobrevivió",
"Sobrevivió")) +
```

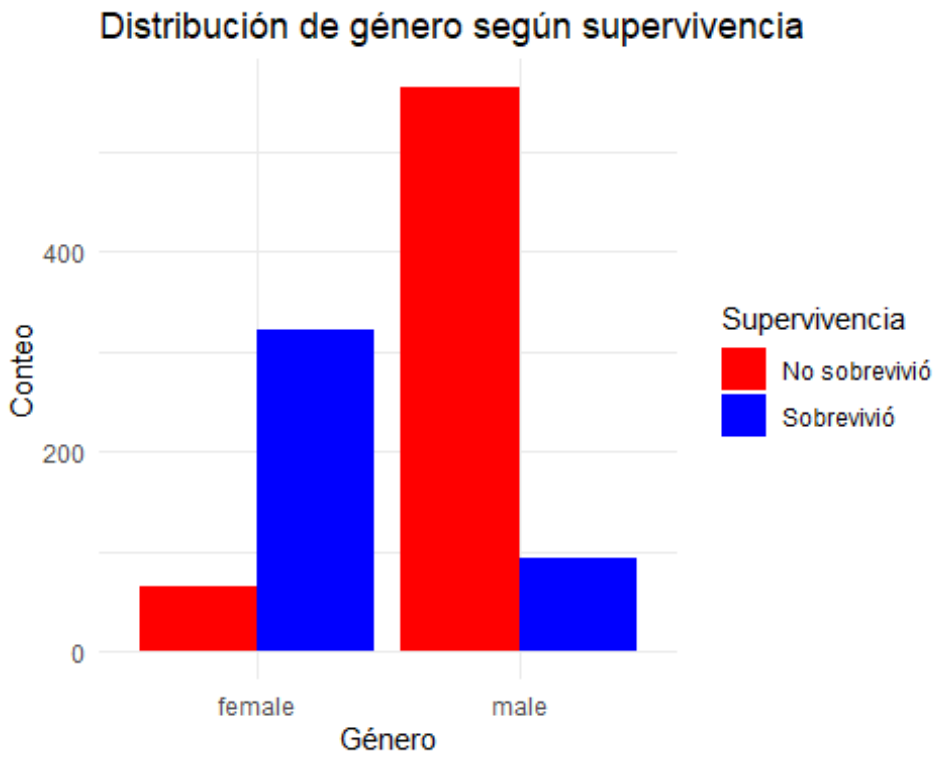
```
  labs(
    title = "Distribución de la edad según supervivencia",
    x = "Edad",
    y = "Frecuencia",
    fill = "Supervivencia"
  ) +
  theme_minimal()
```




```
ggplot(M2, aes(x = factor(Survived), y = Fare, fill = factor(Survived))) +
  geom_boxplot() +
  scale_fill_manual(values = c("red", "blue"), labels = c("No sobrevivió",
"Sobrevivió")) +
  labs(
    title = "Distribución de tarifas según supervivencia",
    x = "Supervivencia",
    y = "Tarifa",
    fill = "Supervivencia"
  ) +
  theme_minimal()
```




```r
ggplot(M2, aes(x = Sex, fill = factor(Survived))) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("red", "blue"), labels = c("No sobrevivió",
"Sobrevivió")) +
  labs(
    title = "Distribución de género según supervivencia",
    x = "Género",
    y = "Conteo",
    fill = "Supervivencia"
  ) +
  theme_minimal()
```



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

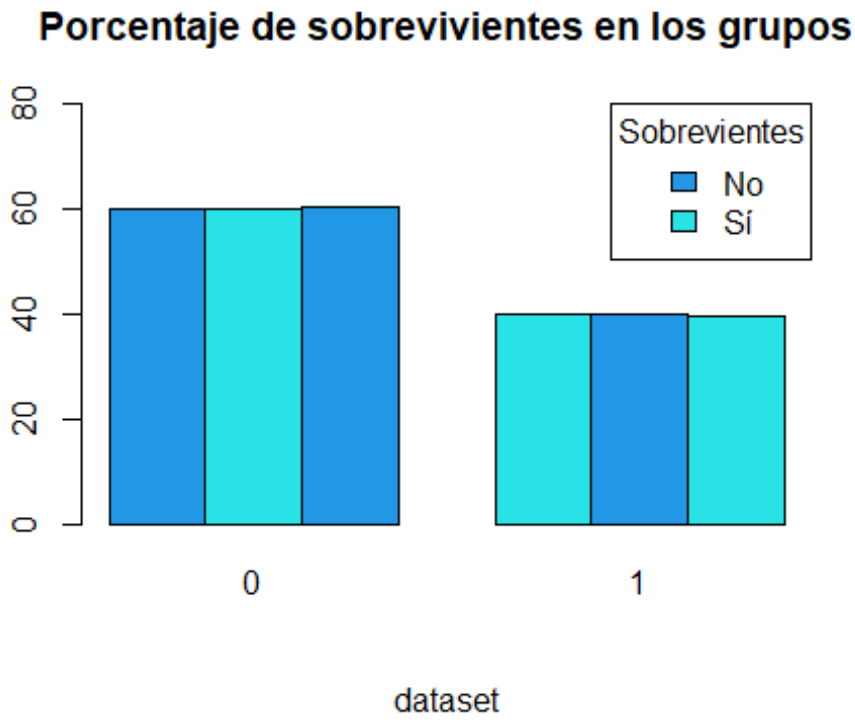
```
# completa
prop_completa <- prop.table(table(M2$Survived))
# entrenamiento
prop_train <- prop.table(table(M_train$Survived))
# validación
prop_valid <- prop.table(table(M_valid$Survived))
# tabla comparativa
TablaComparativa <- rbind(
  Completa = round(prop_completa * 100, 2),
  Entrenamiento = round(prop_train * 100, 2),
  Prueba = round(prop_valid * 100, 2)
)
```

```
print(TablaComparativa)
```

```
##                    0     1
## Completa       60.21 39.79
## Entrenamiento 60.19 39.81
## Prueba         60.26 39.74
```

- Haz un gráfico de barras que te ayude a comparar las tres bases de datos. Auxíliate del código:

```
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)
```



Como podemos observar si se mantuvo la proporción de sobrevivientes en cada subconjunto de la base de datos.

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 ~.-PassengerId, data = M_train, family = "binomial")

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

```
## Start:  AIC=582.91
## Survived ~ (PassengerId + Pclass + Sex + Age + SibSp + Parch +
##     Fare + Embarked) - PassengerId
##
##            Df Deviance    AIC
## - Embarked  2   564.39 580.39
## - Fare      1   563.09 581.09
## - Parch     1   564.51 582.51
## <none>          562.91 582.91
## - SibSp     1   567.68 585.68
## - Age       1   580.31 598.31
## - Pclass    2   596.11 612.11
## - Sex       1   885.30 903.30
##
## Step:  AIC=580.39
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare
##
##            Df Deviance    AIC
## - Fare      1   564.72 578.72
## - Parch     1   565.96 579.96
## <none>          564.39 580.39
## + Embarked  2   562.91 582.91
## - SibSp     1   569.03 583.03
## - Age       1   582.61 596.61
## - Pclass    2   601.67 613.67
## - Sex       1   888.81 902.81
##
## Step:  AIC=578.72
## Survived ~ Pclass + Sex + Age + SibSp + Parch
##
##            Df Deviance    AIC
## - Parch     1   566.02 578.02
## <none>          564.72 578.72
## + Fare      1   564.39 580.39
## + Embarked  2   563.09 581.09
## - SibSp     1   569.21 581.21
## - Age       1   583.35 595.35
## - Pclass    2   628.56 638.56
## - Sex       1   892.66 904.66
##
## Step:  AIC=578.02
## Survived ~ Pclass + Sex + Age + SibSp
```

```
##
##             Df Deviance    AIC
## <none>           566.02 578.02
## + Parch      1   564.72 578.72
## + Fare       1   565.96 579.96
## + Embarked   2   564.52 580.52
## - SibSp      1   572.78 582.78
## - Age        1   584.58 594.58
## - Pclass     2   629.75 637.75
## - Sex        1   899.00 909.00

##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family =
"binomial",
##     data = M_train)
##
## Coefficients:
## (Intercept)      Pclass2      Pclass3      Sexmale          Age
SibSp
##     4.37619     -1.24670     -2.26263     -3.58616     -0.03714     -
0.32325
##
## Degrees of Freedom: 730 Total (i.e. Null);  725 Residual
## Null Deviance:       982.8
## Residual Deviance: 566   AIC: 578
```

- 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 + Fare, family =
"binomial", data = M_train)
summary(B)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Fare, family =
"binomial",
##     data = M_train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.3118795  0.5487947   7.857 3.93e-15 ***
## Pclass2     -1.2042255  0.3442358  -3.498 0.000468 ***
```

```
## Pclass3     -2.2110854  0.3629367  -6.092 1.11e-09 ***
## Sexmale     -3.5798855  0.2430175 -14.731  < 2e-16 ***
## Age         -0.0369697  0.0088928  -4.157 3.22e-05 ***
## SibSp       -0.3276602  0.1289489  -2.541 0.011053 *
## Fare         0.0006345  0.0024736   0.257 0.797560
## ---
## 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: 565.96  on 724  degrees of freedom
## AIC: 579.96
##
## Number of Fisher Scoring iterations: 5
```

Este modelo incluye las variables de clase, genero, edad, hermanos y fare. Como podemos ver tiene una significancia de 560, y todas las variables son bastante significativas, excepto fare que como podemos observar no tiene ningun grado de significancia.

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)  4.376195   0.489185   8.946  < 2e-16 ***
## Pclass2     -1.246699   0.301927  -4.129 3.64e-05 ***
## Pclass3     -2.262629   0.302650  -7.476 7.66e-14 ***
## Sexmale     -3.586162   0.241910 -14.824  < 2e-16 ***
## Age         -0.037142   0.008871  -4.187 2.83e-05 ***
## SibSp       -0.323253   0.127840  -2.529   0.0115 *
## ---
## 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: 566.02  on 725  degrees of freedom
```

```
## AIC: 578.02
##
## Number of Fisher Scoring iterations: 5
```

Este modelo tiene un mejor AIC de 558.6 lo que apunta a una mejor significancia global, además elimina Fare la variable menos importante del modelo anterior lo que lleva a que sus demás variables sean aun más significativas.

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

```
TablaModelos <- data.frame(
  Modelo = c("B", "C"),
  AIC = c(B$aic, C$aic),
  Deviance = c(B$deviance, C$deviance),
  Null_Deviance = c(B$null.deviance, C$null.deviance)
)

TablaModelos
```

Modelo	AIC	Deviance	Null_Deviance
B	579.9558	565.9558	982.7966
C	578.0220	566.0220	982.7966

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

Se mantiene igual ya que el Null_Deviance calcula el ajuste solo por el intercepto y no toma en cuenta las variables.

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

El AIC baja ya que esta medida toma en cuenta la complejidad, al quitar Fare el modelo tiene un mejor resultado. Deviance muestra que ambos modelos mejoran significativamente en comparación al nulo, no hay mucha diferencia ya que Fare no es una variable importante para estos dos 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

```
# Calcular pseudo R^2 para cada modelo
pseudo_r2_B <- 1 - (B$deviance / B$null.deviance)
pseudo_r2_C <- 1 - (C$deviance / C$null.deviance)

# Crear una tabla comparativa
TablaPseudoR2 <- data.frame(
  Modelo = c("B", "C"),
  Pseudo_R2 = c(pseudo_r2_B, pseudo_r2_C)
)

# Mostrar la tabla
TablaPseudoR2
```

Modelo	Pseudo_R2
B	0.4241374
C	0.4240700

Podemos ver que en ambos modelos explican el 44% de la variabilidad, esto contribuye a la hipóstesis de que Fare no impacta en el modelo.

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.

```
Diferencia = C$null.deviance-B$deviance
gl = C$df.null - C$df.deviance

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

Diferencia

## [1] 416.8408

gl

## integer(0)

v
```

```
## numeric(0)
```

Interpreta en el contexto del problema

Como podemos ver el valor-p es extremadamente pequeño ya que la diferencia entre el modelo nulo y el modelo C es de 436. Podemos rechazar H1 y concluimos que el modelo con predictores explica significativamente mejor la probabilidad de supervivencia que el modelo nulo.

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	565.9558	NA	NA	NA
725	566.0220	-1	-0.0662486	0.7968792

Modelo Seleccionado

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

```
c0 = round(C$coefficients[1],3)
c1 = round(C$coefficients[2],3)
c2 = round(C$coefficients[3],3)
c3 = round(C$coefficients[4],3)
c4 = round(C$coefficients[5],3)
c5 = round(C$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:

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

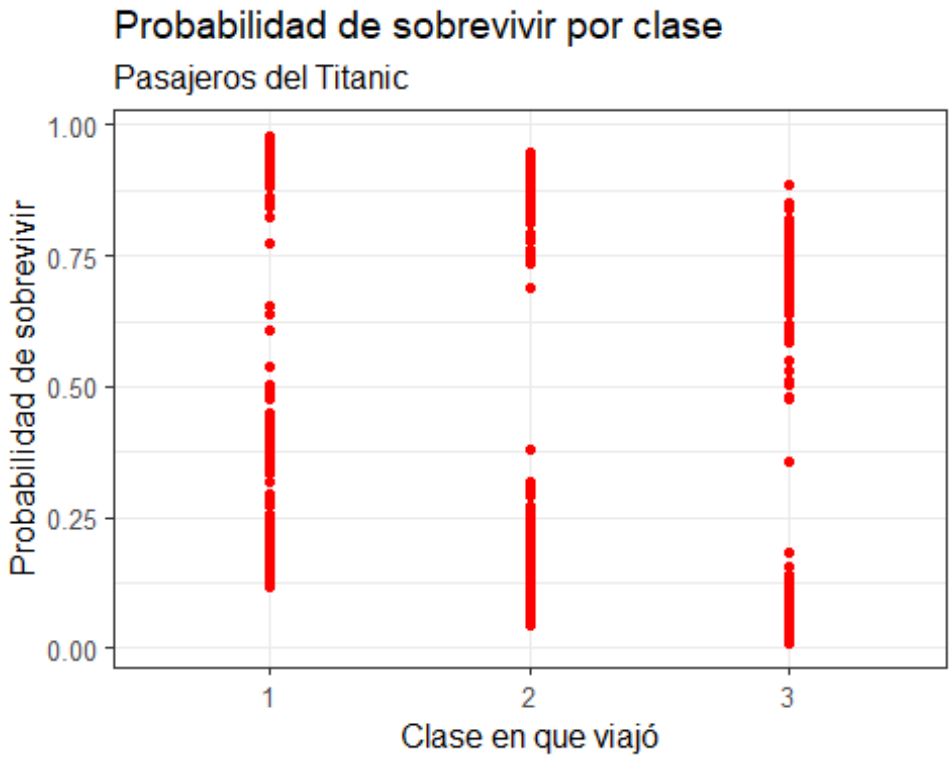
Así que interpretar el efecto de cada variable, se grafica cada una de ellas contra los valores predichos para la probabilidad de sobrevivencia.

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.
```




```
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)
```

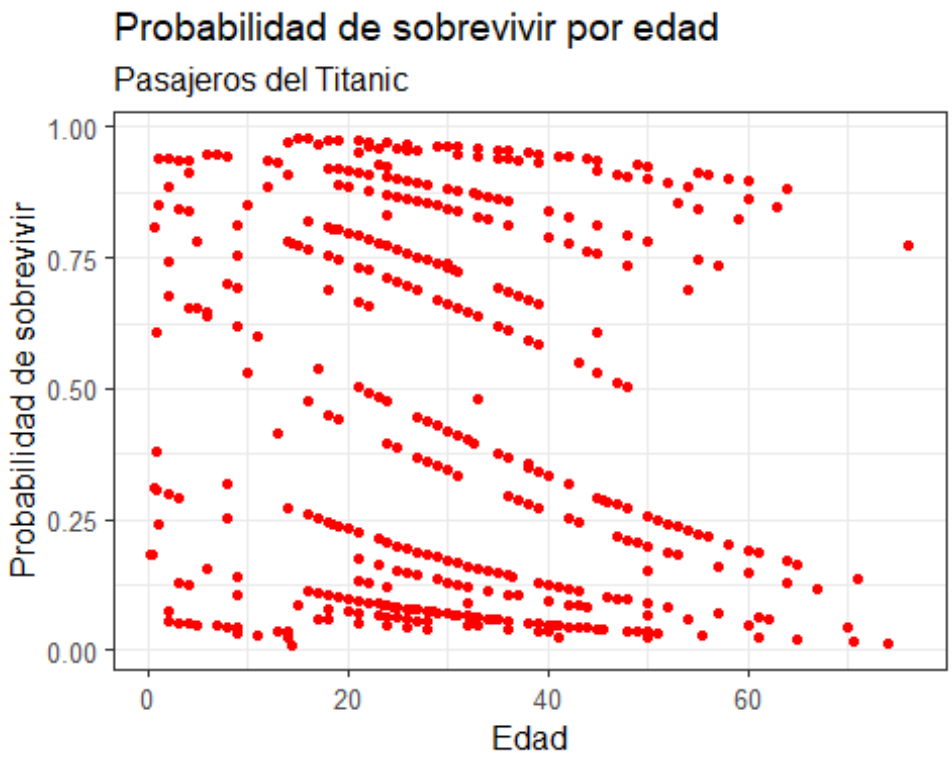
```
## Warning: Use of `M_pred$p_pred` is discouraged.
## ℹ Use `p_pred` instead.
```


```
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.
```


```
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.
```



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

Pclass: Aumentar la clase reduce las probabilidades de sobrevivir (efecto negativo, especialmente en tercera clase). Sex: Ser hombre reduce significativamente la probabilidad de supervivencia. Age: Incrementos en la edad reducen ligeramente la probabilidad de supervivencia. SibSp: Tener más familiares a bordo reduce la probabilidad de sobrevivir.

Predicciones

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

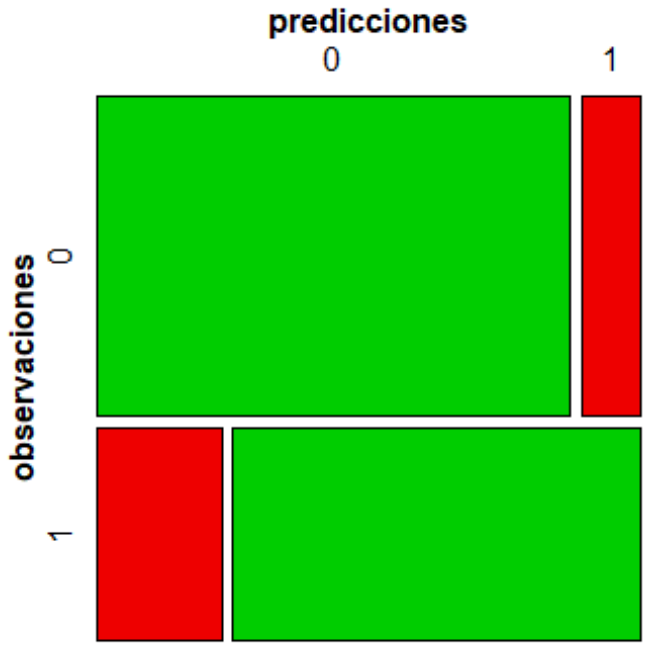
Matriz de confusión

```
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	392	48
1	68	223

```
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 0.8413133

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

## La Sensibilidad del modelo es 0.8909091

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

## La Especificidad del modelo es 0.766323

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

```
## La Precisión del modelo es 0.8521739
```

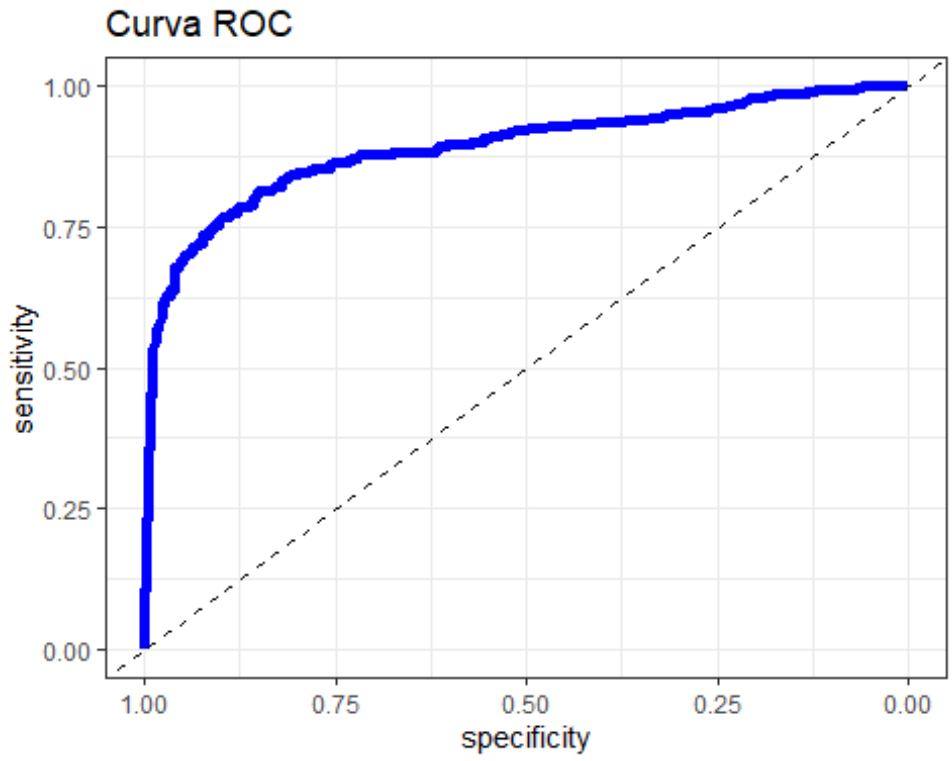
Define si el modelo es bueno o no.

Si es un buen modelo, podemos ver que las predicciones son muy acertadas con la realidad. Además un accuracy de 85% es bastante alto.

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)

## Warning: package 'pROC' was built under R version 4.4.2

## 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: 0.8908

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*

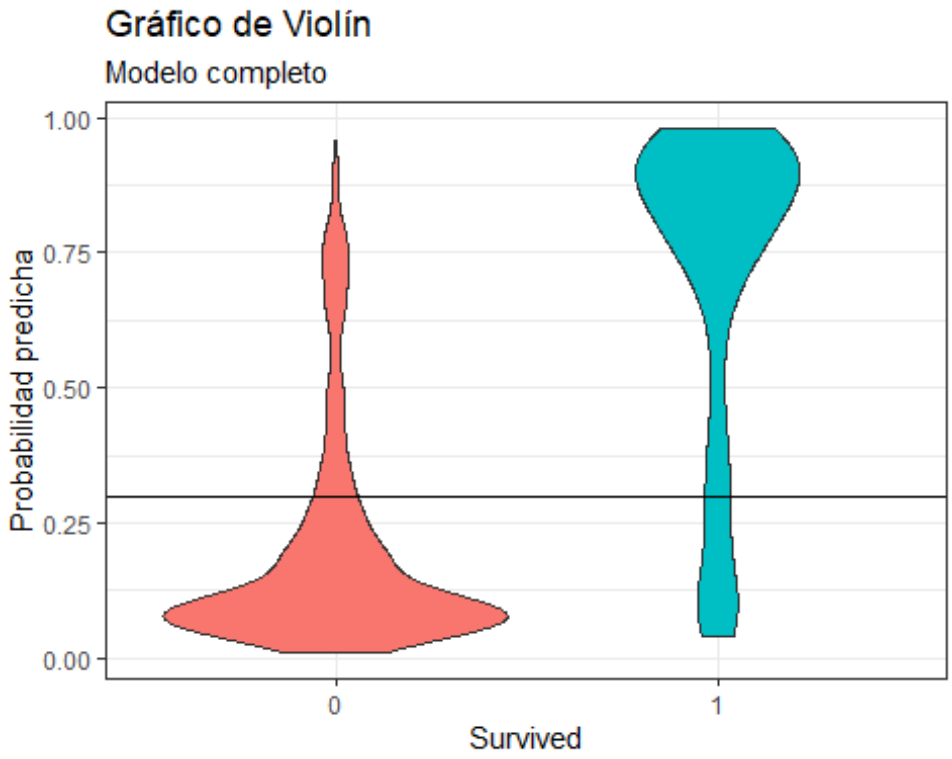
La gráfica indica que el modelo es bueno para diferenciar entre clases, esto significa que es bueno para predecir si una persona sobrevivió o no.

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

Podemos ver que las probabilidades predichas para cada clase estan concentradas en los extremos (cercanas a 0 para los no sobrevivientes y cercanas a 1 para los sobrevivientes), lo que nos indica que el modelo tiene confianza en la mayoría de sus predicciones.

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')
```

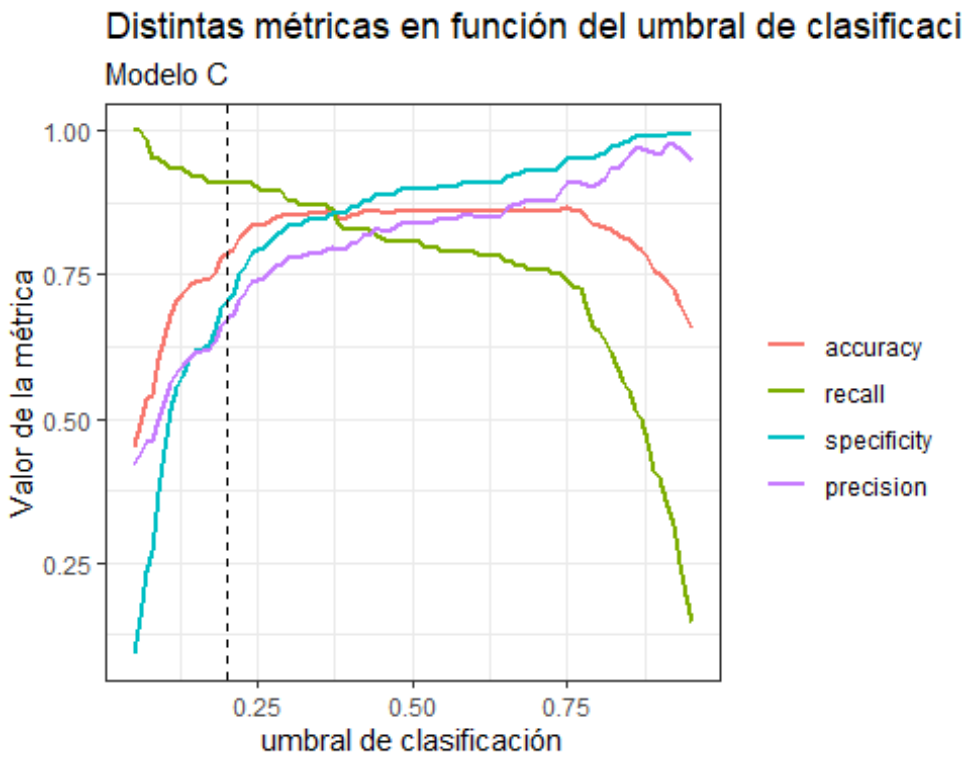
Gráfica

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 = 0.20 #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.

Podemos ver que entre 0.4 y 0.5 el umbral tiene las mejores métricas, esto es antes de que el recall comience a disminuir.

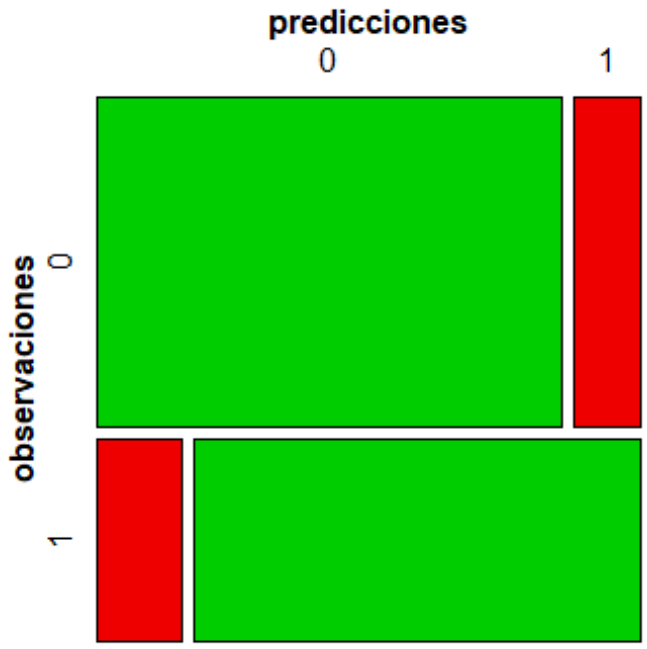
Matriz de confusión con el umbral de clasificación optimo

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 > 0.5, yes = 1, no = 0)
M_Cv <- table(prediccionesV, M_valid$Survived, dnn = c("observaciones",
"predicciones"))
M_Cv
```

observaciones/predicciones	0	1
0	169	24
1	19	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 0.8621795

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

## La Sensibilidad del modelo es 0.8756477
```

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

## La Especificidad del modelo es 0.8403361

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

## La Precisión del modelo es 0.8989362
```

El mejor umbral practicamente es 0.5, este resulta con resultados mayor al 80% en todas las metricas.

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.

```
M_test=read.csv("Titanic_test.csv")
M_test$Pclass <- as.factor(M_test$Pclass)
M_test$Sex <- as.factor(M_test$Sex)
M_test$Survived <- predict(B, newdata = M_test, type = "response")
M_test$Survived <- ifelse(M_test$Survived > 0.5, 1, 0)



cat("Sobrevivio: ", length(which(M_test$Survived==0)) ,"\n")

## Sobrevivio:  203

cat("No Sobrevivio: ", length(which(M_test$Survived==1)) )

## No Sobrevivio:  128

head(M_test,5)
```

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292		Q	0
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000		S	1

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875		Q	0
895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625		S	0
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875		S	1

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.

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

Encontramos que el modelo que mejor se ajusta a los datos proporcionados es el siguiente:

Survived = 4.207061 -1.14(Pclass2) -2.04(Pclass3) - 3.66(Sexmale) -0.03(Age) -0.40(SibSp)

A partir del análisis del modelo podemos ver que hay 4 caracteristicas que determinan la supervivencia de los pasajeros.

Sexo: Ser mujer incrementó significativamente las probabilidades de sobrevivir el Titanic, probablemente por la mentalidad “mujeres y niños primero”. Edad: Las personas más jóvenes tenían una mayor probabilidad de sobrevivir, probablemente debido a la prioridad otorgada a niños para subir a botes salvavidas. Clase de pasaje (Pclass): Pasajeros en primera clase (Pclass 1) tuvieron mayor probabilidad de sobrevivir que aquellos en clases más bajas, probablemente porque estaban ubicados en zonas inferiores y tenian menos acceso a botes en la evacuación. Tamaño del grupo familiar (SibSp): Un tamaño muy grande podría haber disminuido las probabilidades de sobrevivencia. Esta variable es la menos significante.

El umbral optimo es de 0.5 esto mantiene el accuracy alto mientras que recall y precision estan consideradamente balanceados. En casos donde queremos evitar no identificar sobrevivientes (falsos positivos) podemos usar un umbral más bajo a 0.5, esto nos permite tener una mayor precisión y sensibilidad.