

# Actividad Integradora 2

2024-11-19

Utiliza los archivos del Titanic para detectar cuáles fueron las principales características que de las personas que sobrevivieron y elabora en modelo de predicción de sobrevivencia o no en el Titanic. Utiliza en las siguientes bases de datos:

Base de datos del Titanic: Titanic Base de datos de prueba: Titanic\_test

```
datas = read.csv('Titanic.csv')
data_test = read.csv('Titanic_test.csv')
head(datas)
```

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

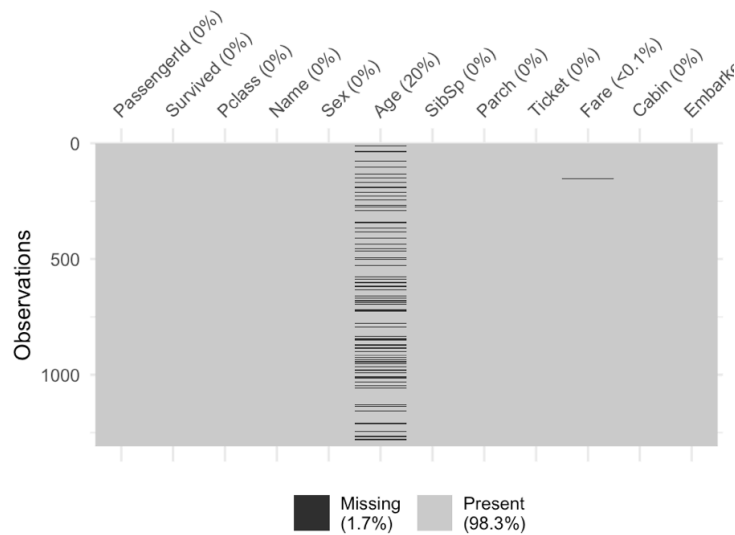
Las variables para la base de datos son las siguientes (excluye aquellas que no sean de interés para el análisis):

- 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)

## 1. Prepara la base de datos Titanic:

- Analiza los datos faltantes

```
library(visdat)
vis_miss(datas)
```



Como podemos observar, la mayoría de las variables no cuenta con datos faltantes, a excepción de ‘Age’, ‘Fare’ y ‘Enmarked’. Además, notemos que en ‘Age’ falta el 20% de los datos.

Dado que no haremos una imputación de los datos, se hará un análisis para visualizar el efecto en las variables categóricas de eliminar todos los registros con valores faltantes.

```
data = na.omit(datas)
dim(data)
```

```
## [1] 1043 12
```

Análisis de Survived

```
t2c = 100*prop.table(table(datas[,2]))
t2s = 100*prop.table(table(data[,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

Al eliminar los registros con valores faltantes, se pierde más información de los muertos (0.97) que de los supervivientes (1.05). Esto podría introducir un sesgo si los datos faltantes no son aleatorios, pero en nuestro caso puede resultar beneficioso, pues ahora tenemos un 60% de muertos y 40% de supervivientes, es decir, podemos conseguir un equilibrio entre clases. Aunque el efecto de pérdida de información no es tan grande, por lo que podría no ser tan significativo.

Análisis de Pclass

```
t3c = 100*prop.table(table(datas[,3]))
t3s = 100*prop.table(table(data[,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

Al eliminar los pasajeros con información faltante perdemos más información de los pasajeros de tercera clase, mientras que la primera y segunda clase empiezan a abarcar más porcentaje de clase. Nuevamente, podemos llegar a la idea de que las clases parecen estar equilibrándose, aunque la tercera clase sigue siendo mayor a la primera y la segunda, por lo que probablemente exista un pequeño sesgo hacia la clase social baja.

Análisis de Sex

```
t4c = 100*prop.table(table(datas[,5]))
t4s = 100*prop.table(table(data[,5]))
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

Al eliminar los datos faltantes también perdemos un poco de información de los pasajeros hombres, aunque no parece afectar de gran manera, pues las clases siguen balanceadas de la misma manera.

Análisis de Embarked

```
t9c = 100*prop.table(table(datas[,12]))
t9s = 100*prop.table(table(data[,12]))
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

Por último, la variable Embarked se ve un poco más afectada en el puerto de Queenstown, pues si eliminamos las filas con datos faltantes perdemos alrededor de la mitad de la información de pasajeros que embarcaron en Queenstown. Por lo que esta sería la variable más afectada hasta el momento.

Conclusión de datos faltantes

Habiendo analizado los efectos uno por uno, pudimos notar que no hay mucho efecto adverso si eliminamos las filas con valores faltantes, la información que se pierde es muy poca, y en algunos casos nos beneficia que las clases se equilibren un poco más. Por lo que, trabajaremos con la base de datos sin NAs.

- Realiza un análisis descriptivo

```
summary(data)
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Min. : 1.0	Min. :0.0000	Min. :1.000	Length:1043	Length:1043	Min. : 0.17	Min. :0.0000	Min. :0.0000	Length:1043	Min. : 0.00	Length:1043	Length:1043
1st Qu.: 326.5	1st Qu.:0.0000	1st Qu.:1.000	Class :character	Class :character	1st Qu.:21.00	1st Qu.:0.0000	1st Qu.:0.0000	Class :character	1st Qu.: 8.05	Class :character	Class :character
Median : 662.0	Median :0.0000	Median :2.000	Mode :character	Mode :character	Median :28.00	Median :0.0000	Median :0.0000	Mode :character	Median : 15.75	Mode :character	Mode :character
Mean : 655.4	Mean :0.3979	Mean :2.209	NA	NA	Mean :29.81	Mean :0.5043	Mean :0.4219	NA	Mean : 36.60	NA	NA
3rd Qu.: 973.5	3rd Qu.:1.0000	3rd Qu.:3.000	NA	NA	3rd Qu.:39.00	3rd Qu.:1.0000	3rd Qu.:1.0000	NA	3rd Qu.: 35.08	NA	NA

Max. :1307.0	Max. :1.0000	Max. :3.000	NA	NA	Max. :80.00	Max. :8.0000	Max. :6.0000	NA	Max. :512.33	NA	NA
--------------	--------------	-------------	----	----	-------------	--------------	--------------	----	--------------	----	----

```
sapply(data, sd)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
```

```
## PassengerId  Survived  Pclass     Name    Sex     Age
## 377.5270355   0.4896975   0.8406853   NA      NA    14.3662545
## SibSp       Parch    Ticket     Fare    Cabin  Embarked
## 0.9130797    0.8406546    NA      55.7536477   NA      NA
```

La mayoría de las variables están completas y tienen distribuciones razonables, excepto algunas (Cabin, Embarked, Age) con valores faltantes significativos.

- Haz una partición de los datos (70-30) para el entrenamiento y la validación. Revisa la proporción de sobrevivientes para la partición y la base original.

```
library(creditmodel)
```

```
## Package 'creditmodel' version 1.3.1
```

```
df = train_test_split(data, prop = 0.7, split_type = "Random", seed = 43)
dim(df$train)
```

```
## [1] 730 12
```

```
dim(df$test)
```

```
## [1] 313 12
```

2. Con la base de datos de entrenamiento, encuentra un modelo logístico para encontrar el mejor conjunto de predictores que auxilien a clasificar la dirección de cada observación.

- Auxiliarte del criterio de AIC para determinar cuál es el mejor modelo.

El código se encuentra comentado por el tiempo que tarda en cargar y la cantidad de modelos que arroja. Pero se recopilieron los 2 mejores después de correrlo, se presentarán más adelante.

```
#model = glm(Survived ~ ., data = df$train, family = "binomial")
#step(model, direction="both", trace=1)
```

El mejor modelo que predice la variable "Survived" es aquel que considera la edad, sexo, clase y hermanos de los pasajeros, con un AIC de 569.85

- Propón por lo menos los dos que consideres mejores modelos.

Modelo 1: Modelo logístico que predice la supervivencia de un pasajero con base en su edad, sexo, clase social y hermanos a bordo.

```
model2 = glm(Survived ~ Age+Sex+Pclass+SibSp, data = df$train, family = "binomial")
summary(model2)
```

```
##
## Call:
## glm(formula = Survived ~ Age + Sex + Pclass + SibSp, family = "binomial",
##      data = df$train)
##
## Coefficients:
##              Estimate Std. Error z value      Pr(>|z|)
## (Intercept)  4.970063   0.562589   8.834 < 0.0000000000000002 ***
## Age         -0.032370   0.008325  -3.888   0.000101 ***
## Sexmale     -3.641294   0.237904 -15.306 < 0.0000000000000002 ***
## Pclass      -0.969514   0.148547  -6.527   0.000000000000673 ***
## SibSp       -0.362526   0.126596  -2.864   0.004188 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 984.20  on 729  degrees of freedom
## Residual deviance: 559.85  on 725  degrees of freedom
## AIC: 569.85
##
## Number of Fisher Scoring iterations: 5
```

Modelo 2: Modelo logístico que predice la supervivencia de un pasajero con base en edad, sexo, clase social, hermanos a bordo y su ID.

```
model3 = glm(Survived ~ Age+Sex+Pclass+SibSp+PassengerId, data = df$train, family = "binomial")
summary(model3)
```

```
##
## Call:
## glm(formula = Survived ~ Age + Sex + Pclass + SibSp + PassengerId,
##      family = "binomial", data = df$train)
##
## Coefficients:
##              Estimate Std. Error z value      Pr(>|z|)
## (Intercept)  5.1966442  0.6183928   8.403 < 0.0000000000000002 ***
## Age         -0.0326751  0.0083212  -3.927   0.000086114296 ***
## Sexmale     -3.6418223  0.2382499 -15.286 < 0.0000000000000002 ***
## Pclass      -0.9854261  0.1498971  -6.574   0.00000000000049 ***
## SibSp       -0.3715224  0.1273724  -2.917   0.00354 **
## PassengerId -0.0002707  0.0002937  -0.922   0.35673
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 984.2  on 729  degrees of freedom
## Residual deviance: 559.0  on 724  degrees of freedom
## AIC: 571
##
## Number of Fisher Scoring iterations: 5
```

Estos dos modelos son los de mejor AIC obtenido en step.

### 3. Analiza los modelos a través de:

#### - Identificación de la Desviación residual de cada modelo

Modelo 1: 559.85

Modelo 2: 559

Aunque la diferencia en la desviación residual entre ambos modelos es pequeña, Modelo 2 muestra un ajuste ligeramente mejor que Modelo 1 debido a su desviación residual más baja. Sin embargo, la magnitud de esta diferencia puede no ser lo suficientemente significativa y es importante considerar otras métricas de desempeño y características del modelo para tomar una decisión definitiva sobre cuál es superior.

#### - Identificación de la Desviación nula

Modelo 1: 984.2

Modelo 2: 984.2

Ambos modelos tienen la misma desviación nula, lo que indica que comienzan con un nivel de error de referencia idéntico antes de considerar los predictores. Dado que la desviación nula es igual, cualquier mejora en la desviación residual será clave para determinar la calidad de ajuste de cada modelo.

#### - Cálculo de la Desviación Explicada

```
r2_model2 = 1 - (559.85 / 984.2)
r2_model3 = 1 - (559 / 984.2)

cat('La pseudo r cuadrada del primer modelo es ', r2_model2, '\n')
```

```
## La pseudo r cuadrada del primer modelo es 0.4311624
```

```
cat('La pseudo r cuadrada del primer modelo es ', r2_model3)
```

```
## La pseudo r cuadrada del primer modelo es 0.432026
```

El modelo 2 tiene mayor pseudo r cuadrada, aunque por muy poca diferencia al modelo 1, esto indica que el Modelo 2 explica un poco más de la variabilidad de los datos en comparación con Modelo 1, pero solo por un 0.001%.

## - Prueba de la razón de verosimilitud

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

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

Se calcula el estadístico de  $\chi^2$  para la razón de verosimilitud a partir de las *Deviance* de los modelos.

```
Diferencia = model2$null.deviance-model2$deviance
gl = model2$df.null - model2$df.deviance

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

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

Dado el p-value obtenido y suponiendo un nivel de significancia de  $\alpha = 0.05$ , entonces podemos decir que no contamos con suficiente evidencia para rechazar la hipótesis inicial, por lo que el modelo 1 logístico explica mejor la supervivencia de los pasajeros del Titanic que el modelo nulo.

```
Diferencia = model3$null.deviance-model3$deviance
gl = model3$df.null - model3$df.deviance

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

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

Dado el p-value obtenido y suponiendo un nivel de significancia de  $\alpha = 0.05$ , entonces podemos decir que no contamos con suficiente evidencia para rechazar la hipótesis inicial, por lo que el modelo 2 logístico explica mejor la supervivencia de los pasajeros del Titanic que el modelo nulo.

## - Define cuál es el mejor modelo

Dado la similitud de ambos modelos en cuanto a desviación residual, nula, explicada y el resultado de la prueba de razón de verosimilitud, podemos decir que el mejor modelo es el 1, aquel que considera la edad, sexo, clase y hermanos del pasajero. En este caso, la decisión se basó en que aunque el modelo 2 explica un 0.001% más que el modelo 1, lo logra a causa de tomar en consideración una variable adicional, que es el número de registro del pasajero.

## - Escribe su ecuación, analiza sus coeficientes y detecta el efecto de cada predictor en la clasificación.

El modelo logístico que predice la supervivencia de un pasajero en el Titanic es:

$$P(\text{Survived}) = 4.97 - 0.0323 \times \text{Age} - 3.6412 \times \text{SexMale} - 0.9695 \times \text{Pclass} - 0.3625 \times \text{SibSp}$$

El modelo predice la probabilidad de supervivencia en función de varias variables, cuyos coeficientes podemos interpretar cómo:

- Age: A mayor edad, menor es la probabilidad de supervivencia.
- Sex: Ser hombre reduce considerablemente la probabilidad de supervivencia en comparación con ser mujer.
- Pclass: Viajar en una clase más baja reduce la probabilidad de supervivencia.
- SibSp: Tener más hermanos/cónyuges a bordo está asociado con una menor probabilidad de supervivencia.

En resumen, las personas más jóvenes, las mujeres, las que viajan en clases más altas y aquellas con menos familiares a bordo tienen mayores probabilidades de sobrevivir.

Prueba de significancia:

- $H_0$  :  $\beta_i = 0$ . La variable no es significativa para el modelo.
- $H_1$  :  $\exists \beta_i \neq 0$ . La variable es significativa para el modelo.

Dados los valores p de todas los coeficientes del modelo 1 y suponiendo un nivel de significancia de  $\alpha = 0.05$ , podemos decir entonces que contamos con suficiente evidencia para rechazar la hipótesis inicial, por lo que todos los coeficientes del modelo 1 son significantes para el modelo, es decir, la edad, sexo, clase y hermanos del pasajero son importantes para predecir su supervivencia.

4. Analiza las predicciones para los datos de entrenamiento

- Elabora la matriz de confusión

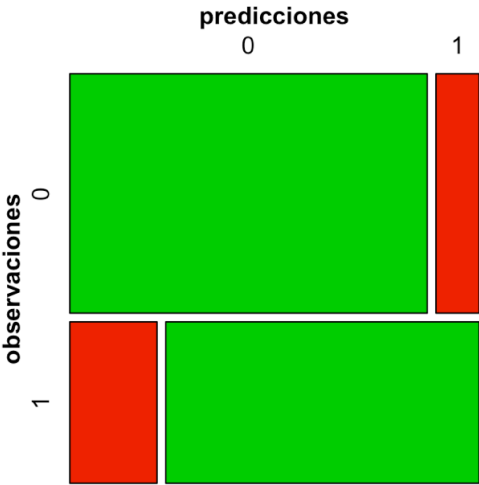
```
library(vcd)

## Loading required package: grid

predicciones <- ifelse(test = model2$fitted.values > 0.5, yes = 1, no = 0)
M_C <- table(model2$model$Survived, predicciones, dnn = c("observaciones", "predicciones"))
M_C

observaciones/predicciones                                0      1
0      389      47
1      64      230

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



El modelo tiene un buen desempeño en la clase 0 (no sobrevivió), dado que una gran parte de las observaciones de esta clase fueron correctamente predichas. Hay más errores en la clase 1 (sobrevivió), ya que los rectángulos rojos en fila 1 y columna 0 son más grandes comparados con los de fila 0 y columna 1. Por lo que el modelo parece adaptarse mejor a los casos de pasajeros que no sobrevivieron.

```
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 0.8479452

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

## La Sensibilidad del modelo es 0.8922018

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

```
## La Especificidad del modelo es 0.7823129
```

```
## La Especificidad del modelo es 0.7823129
```

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

```
## La Precisión del modelo es 0.8587196
```

El modelo funciona mejor para la clase 1 (alta sensibilidad y precisión) que para la clase 0 (menor especificidad). La discrepancia entre sensibilidad y especificidad podría ser un indicativo de un desbalance en las clases o de que el modelo está sesgado hacia una clase específica.

## - Elabora la Curva ROC

```
pred = predict(model2, data = df$train, type = 'response')
library(pROC)
```

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

```
##
## Attaching package: 'pROC'
```

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

```
ROC <- roc(response=df$train$Survived, predictor=pred)
```

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

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

```
ROC
```

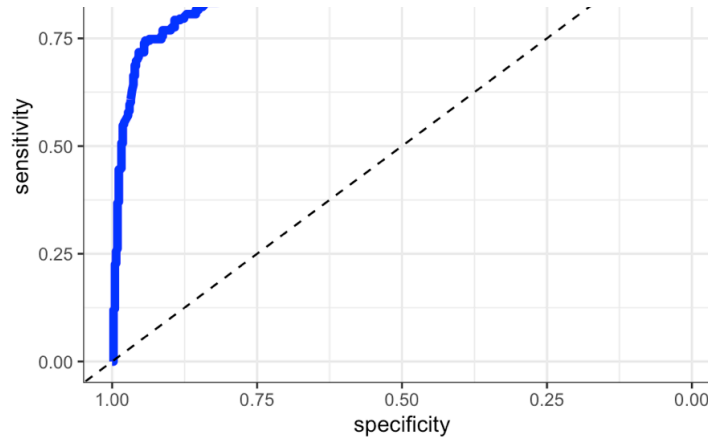
```
##
## Call:
## roc.default(response = df$train$Survived, predictor = pred)
##
## Data: pred in 436 controls (df$train$Survived 0) < 294 cases (df$train$Survived 1).
## Area under the curve: 0.8931
```

El modelo es capaz de discriminar correctamente entre sobrevivientes y no sobrevivientes en el 89.31% de las veces. Es adecuado para el problema de clasificación, con una fuerte capacidad de predicción.

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







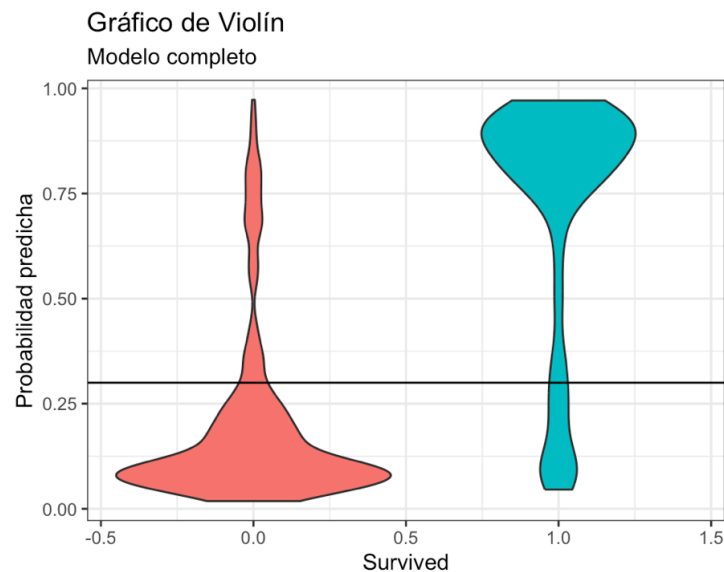
El modelo tiene un buen desempeño global, especialmente para umbrales que mantienen alta sensibilidad con una baja tasa de falsos positivos. Esto lo hace útil si identificar correctamente los casos positivos (sobrevivió) es prioritario.

### - Elabora el gráfico de violín

```
v_d = data.frame(Survived=df$train$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.
```



El modelo tiene un buen desempeño al predecir ambas clases de supervivencia, ya que las distribuciones de probabilidad están bien diferenciadas. Sin embargo, notemos que algunos errores se concentran en los valores cercanos al umbral (probabilidades alrededor de 0.5), lo que sugiere que ajustar el umbral podría mejorar la precisión o recall.

### - Concluye sobre el modelo basándote en las predicciones de los datos de entrenamiento.

El modelo tiene un buen desempeño general, con alta sensibilidad y precisión, y es confiable para predecir la supervivencia en este conjunto de datos. Sin embargo, si se optimiza el umbral se puede maximizar el desempeño en la identificación de sobrevivientes o no sobrevivientes.

## 5. Validación del modelo con la base de datos de validación

## - Elije un umbral de clasificación óptimo

### Generación de base de datos para graficar

```
pred_val = predict(model2, newdata=df$test, type='response')
clase_real = df$test$Survived

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

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

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

### Gráfica

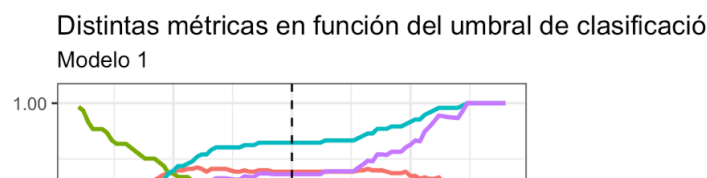
En la gráfica se define cuál es el mejor umbral de clasificación dependiendo de cuál métrica es más importante en el contexto del problema (Exactitud, Sensibilidad, Especificidad o Precisión). Si no hay una métrica de preferencia, se opta por escoger el máximo valor de que pueden tener estas métricas 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.

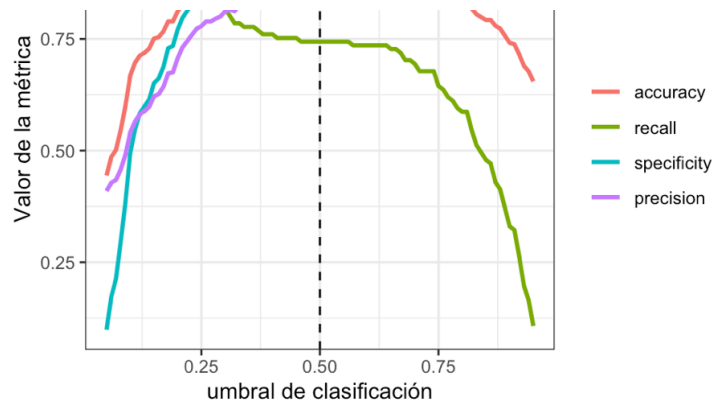
```
library(ggplot2)

u = 0.5 #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 1',
        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.
```





Si se quiere maximizar la precisión (reducir falsos positivos) sin sacrificar demasiado el recall, se puede buscar un punto en el que la línea morada esté relativamente alta sin que el recall (línea verde) caiga drásticamente. En cambio, si se busca maximizar el recall (detectar la mayoría de los positivos) sin demasiadas restricciones de precisión, se debería optar por un umbral más bajo. En términos generales, el umbral alrededor de 0.5 da un modelo estable, ya que proporciona un buen balance de recall, especificidad, y precisión.

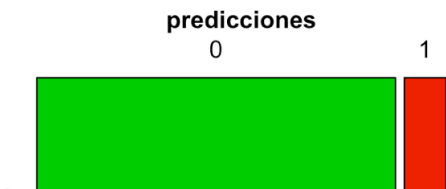
Como nuestro caso es buscar un modelo que identifique si un pasajero sobrevive, es decir, buscamos detectar la mayoría de casos positivos, entonces debemos trabajar con un umbral más bajo. De manera visual, el umbral deseado puede encontrarse entre 0.25 y 0.3 aproximadamente, pero trabajaremos con 0.25.

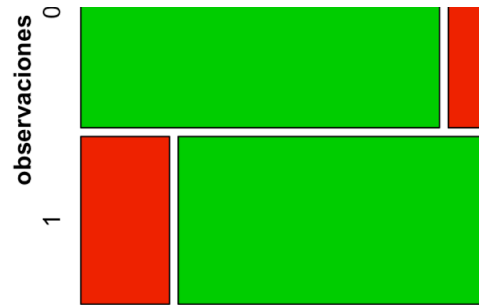
- Elabora la matriz de confusión con el umbral de clasificación óptimo

```
prediccionesV = ifelse(pred_val > 0.25, yes = 1, no = 0)
M_Cv <- table(prediccionesV, df$test$Survived, dnn = c("observaciones", "predicciones"))
M_Cv
```

observaciones/predicciones	0	1
0	163	19
1	29	102

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





La matriz de confusión muestra que el modelo tiene un buen rendimiento general, ya que realiza muchas predicciones correctas. Sin embargo, hay algunas predicciones incorrectas que corresponden a falsos positivos y falsos negativos.

```
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 0.8466454
```

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

```
## La Sensibilidad del modelo es 0.8956044
```

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

```
## La Especificidad del modelo es 0.778626
```

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

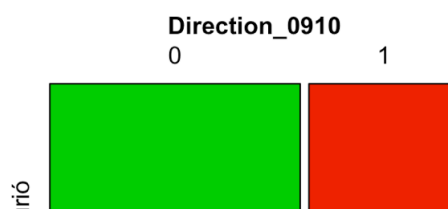
```
## La Precisión del modelo es 0.8489583
```



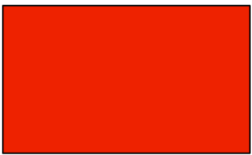
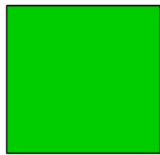
El modelo probado con los datos de prueba y el umbral optimizado tiene un rendimiento más o menos equilibrado entre la detección de positivos (alta sensibilidad) y la clasificación correcta de negativos (alta especificidad), mientras mantiene una precisión elevada en sus predicciones positivas. Esto indica que, en general, realiza predicciones fiables y minimiza tanto los falsos positivos como los falsos negativos.

## 6. Elabora el testeo con la base de datos de prueba.

```
#Cálculo de la probabilidad predicha por el modelo con los datos de test
prob <- predict(model2, newdata = data_test, type = "response")
# Vector de elementos "Down"
pred <- rep("Murió", length(prob))
# Sustitución de "Down" por "Up" si la p > 0.5
pred[prob > 0.25] <- "Sobrevivió"
Direction_0910 = data[1:418,2]
# Matriz de confusión
matriz_confusion <- table(pred, Direction_0910)

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



pred	Mu		
	Sobrevivió		

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

```
## La Exactitud (accuracy) del modelo es 0.5358852
```

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

```
## La Sensibilidad del modelo es 0.625
```

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

```
## La Especificidad del modelo es 0.3831169
```

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

```
## La Precisión del modelo es 0.6346154
```

El modelo cuando se enfrenta a datos sin etiquetar tiene un rendimiento moderado para predecir casos positivos ("Sobrevivió"), pero presenta un problema considerable al clasificar negativos ("Murió"), ya que la especificidad baja y la exactitud global indican una clasificación incorrecta en una proporción alta de estos casos. Esto podría hacer que el modelo no sea adecuado para aplicaciones donde la correcta detección de la clase "Murió" sea crítica.

## 7. Concluye en el contexto del problema:

- Define las principales características que influyen en el modelo seleccionado e interpretalas: ¿qué características tuvieron las personas que sobrevivieron?

- Interpreta los coeficientes del modelo

Como se mencionó en análisis anteriores, las características que más influyen en la supervivencia de un pasajero son la edad, el sexo, la clase social y el número de hermanos a bordo. Y de acuerdo a los coeficientes obtenidos para el modelo, las características de las personas que sobrevivieron probablemente fueron las personas más jóvenes, las mujeres, las que viajaban en clases más altas y aquellas con menos familiares a bordo. Es decir, si se introdujera un nuevo pasajero cuya edad es joven, es mujer, de primera clase y que no tiene hermanos a bordo, es muy probable que sobreviviera.

- Define cuál es el mejor umbral de clasificación y por qué

Como se explicó en análisis anteriores, dado que deseamos maximizar el número de casos positivos, es decir, identificar con éxito la mayoría de todos los casos de supervivencia, entonces debemos trabajar con un umbral más bajo, buscando un equilibrio entre las métricas de especificidad, precisión y exactitud, además de un alto recall. Es preferible, en el caso del Titanic, contar con más falsos positivos que falsos negativos, pues suponer la muerte de un pasajero implica peores consecuencias que suponerlo vivo.