integradora2

2024-11-19

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(VIM)
```

```
## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```r
library(tidyr)
library(caTools)
library(pROC)
```

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

##
## Attaching package: 'pROC'

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

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

```r
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
titanic = read.csv("documents/Titanic.csv", stringsAsFactors = FALSE)
titanic_test = read.csv("documents/Titanic_test.csv", stringsAsFactors = FALSE)

head(titanic)
```

```
##   PassengerId Survived Pclass                                         Name
## 1         892        0      3                             Kelly, Mr. James
## 2         893        1      3             Wilkes, Mrs. James (Ellen Needs)
## 3         894        0      2                    Myles, Mr. Thomas Francis
## 4         895        0      3                             Wirz, Mr. Albert
## 5         896        1      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist)
## 6         897        0      3                   Svensson, Mr. Johan Cervin
##      Sex  Age SibSp Parch  Ticket    Fare Cabin Embarked
## 1   male 34.5     0     0  330911  7.8292              Q
## 2 female 47.0     1     0  363272  7.0000              S
## 3   male 62.0     0     0  240276  9.6875              Q
## 4   male 27.0     0     0  315154  8.6625              S
## 5 female 22.0     1     1 3101298 12.2875              S
## 6   male 14.0     0     0    7538  9.2250              S
```

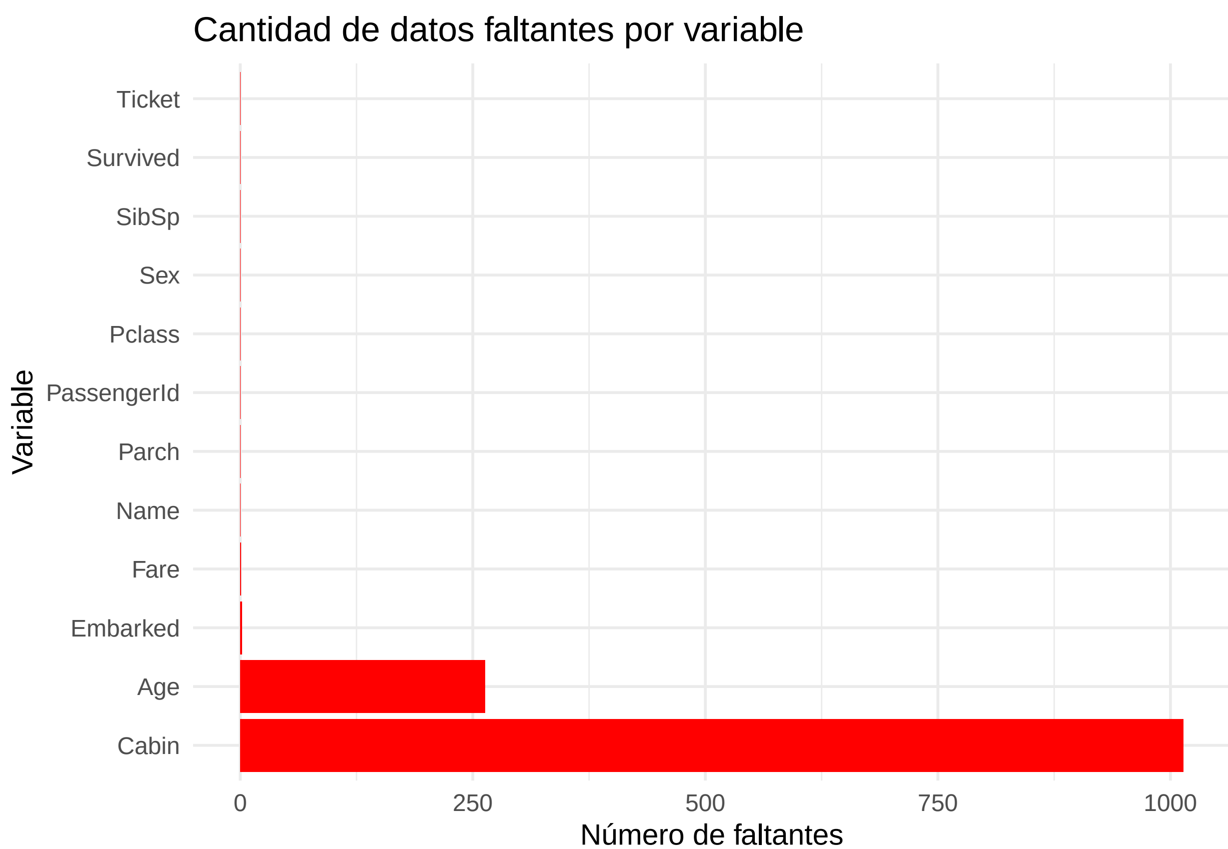
```r
missing_data = colSums(is.na(titanic) | titanic == "")
print("Datos faltantes por columna:")
```

```
## [1] "Datos faltantes por columna:"
```

```r
print(missing_data)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1        1014           2
```

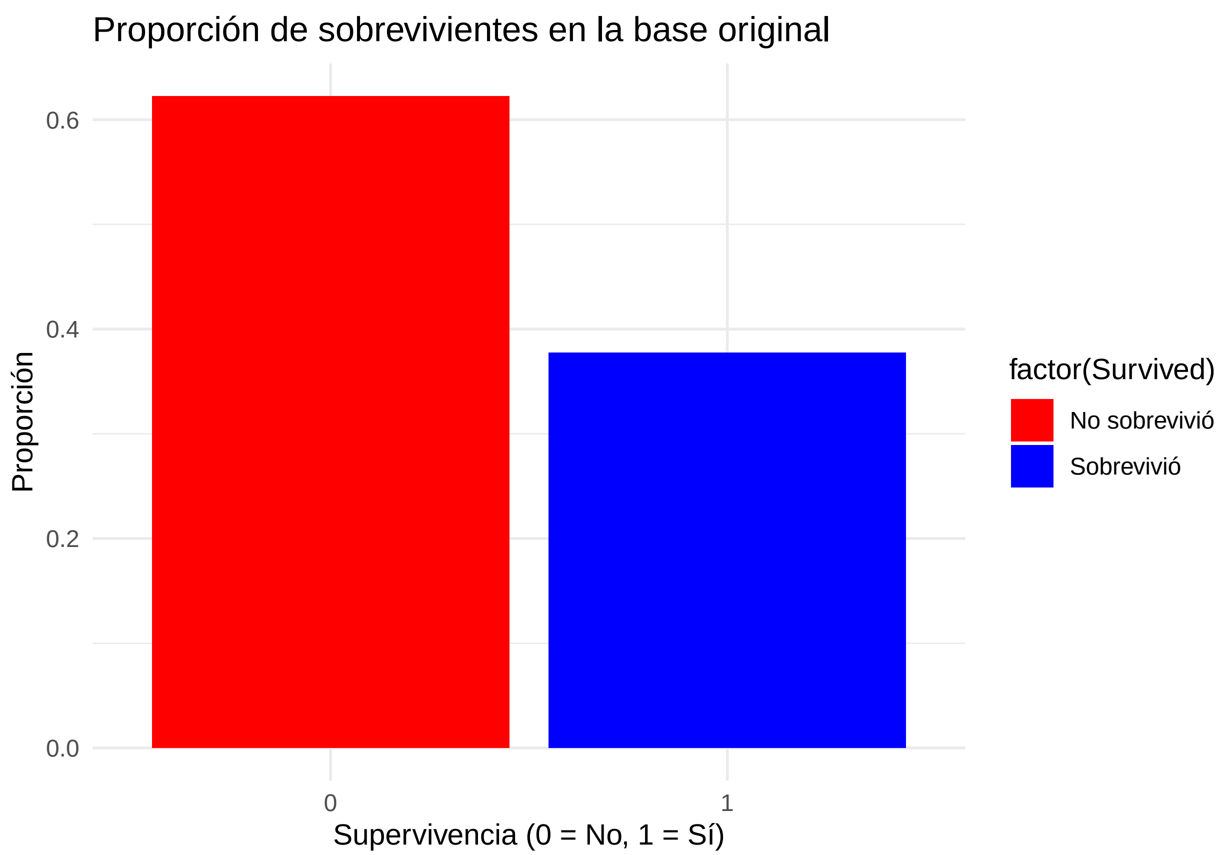
```r
missing_df = titanic %>%
  summarise_all(~ sum(is.na(.) | . == "")) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Faltantes")

ggplot(missing_df, aes(x = reorder(Variable, -Faltantes), y = Faltantes)) +
  geom_bar(stat = "identity", fill = "red") +
  coord_flip() +
  labs(title = "Cantidad de datos faltantes por variable",
       x = "Variable",
       y = "Número de faltantes") +
  theme_minimal()
```




```r
survival_summary = titanic %>%
  group_by(Survived) %>%
  summarise(Count = n()) %>%
  mutate(Proportion = Count / sum(Count))

print(survival_summary)
```

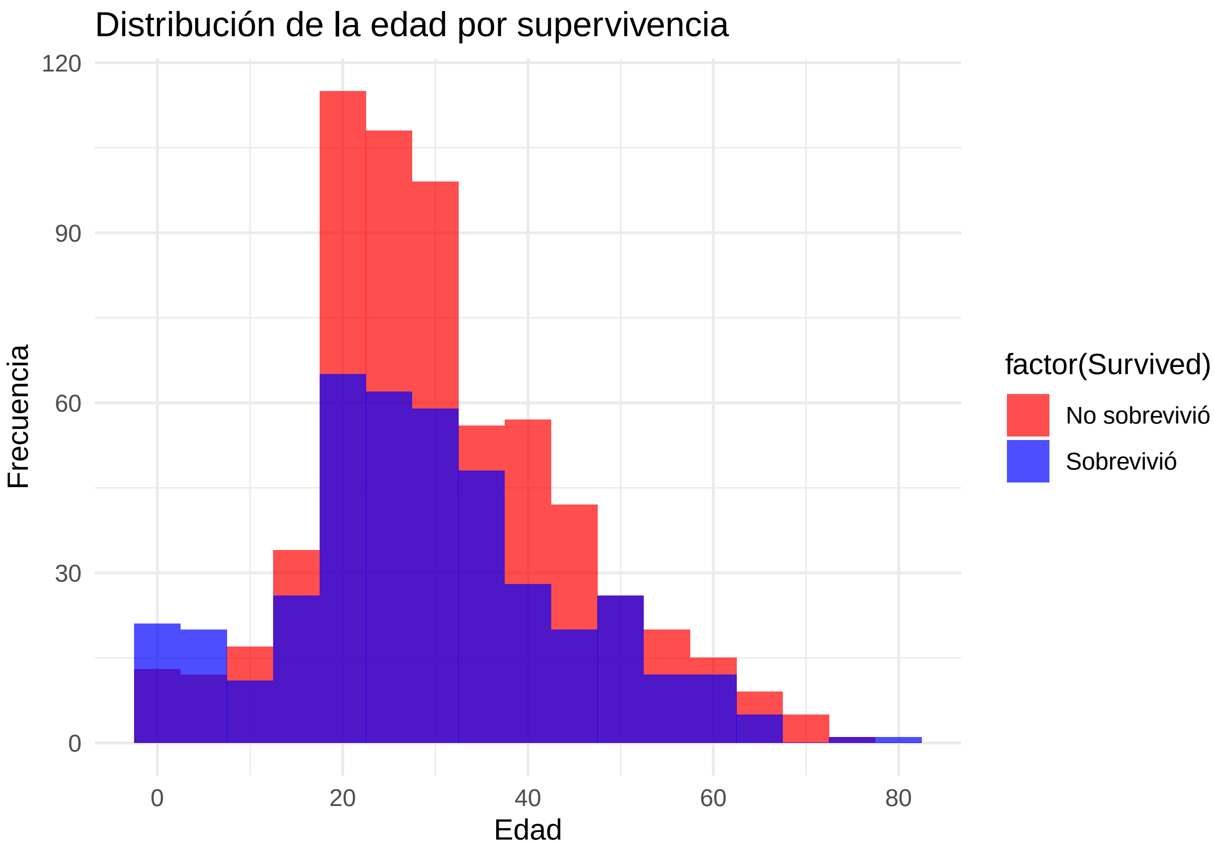
```
## # A tibble: 2 x 3
##   Survived Count Proportion
##      <int> <int>      <dbl>
## 1        0   815      0.623
## 2        1   494      0.377
```

```r
ggplot(survival_summary, aes(x = factor(Survived), y = Proportion, fill = factor(Survived))) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("red", "blue"), labels = c("No sobrevivió", "Sobrevivió")) +
  labs(title = "Proporción de sobrevivientes en la base original",
       x = "Supervivencia (0 = No, 1 = Sí)",
       y = "Proporción") +
  theme_minimal()
```

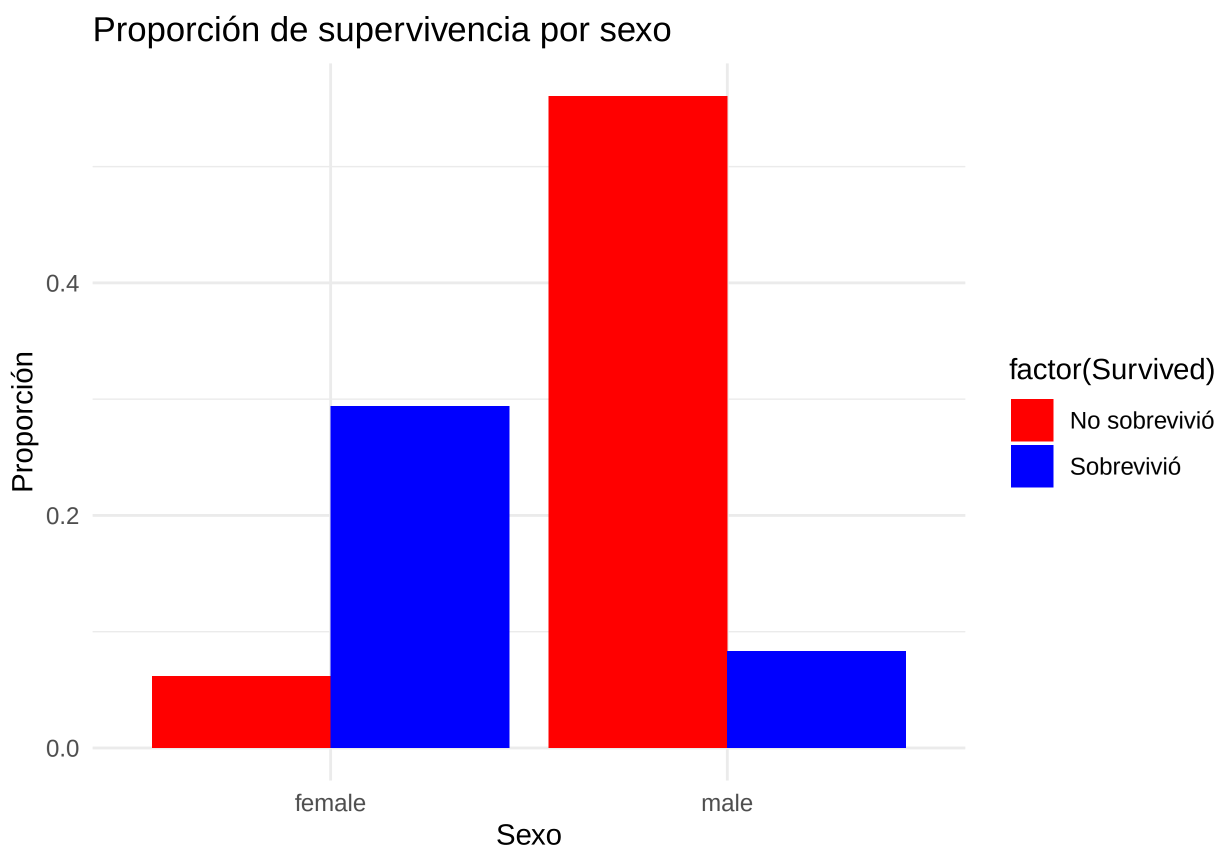



```r
ggplot(titanic, aes(x = Age, fill = factor(Survived))) +
  geom_histogram(binwidth = 5, alpha = 0.7, position = "identity") +
  scale_fill_manual(values = c("red", "blue"), labels = c("No sobrevivió", "Sobrevivió")) +
  labs(title = "Distribución de la edad por supervivencia",
       x = "Edad",
       y = "Frecuencia") +
  theme_minimal()
```

```
## Warning: Removed 263 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
sex_summary = titanic %>%
  group_by(Sex, Survived) %>%
  summarise(Count = n(), .groups = "drop") %>%
  mutate(Proportion = Count / sum(Count))

ggplot(sex_summary, aes(x = Sex, y = Proportion, fill = factor(Survived))) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c("red", "blue"), labels = c("No sobrevivió", "Sobrevivió")) +
  labs(title = "Proporción de supervivencia por sexo",
       x = "Sexo",
       y = "Proporción") +
  theme_minimal()
```




```r
set.seed(123)
split = sample.split(titanic$Survived, SplitRatio = 0.7)
train = subset(titanic, split == TRUE)
test = subset(titanic, split == FALSE)

train_survival = train %>%
  group_by(Survived) %>%
  summarise(Count = n()) %>%
  mutate(Proportion = Count / sum(Count))

test_survival = test %>%
  group_by(Survived) %>%
  summarise(Count = n()) %>%
  mutate(Proportion = Count / sum(Count))

print(train_survival)
```

```
## # A tibble: 2 x 3
##   Survived Count Proportion
##      <int> <int>      <dbl>
## 1        0   570      0.622
## 2        1   346      0.378
```

```r
print(test_survival)
```

```
## # A tibble: 2 x 3
##   Survived Count Proportion
##      <int> <int>      <dbl>
```

```
## 1        0   245      0.623
## 2        1   148      0.377
```

```r
print(survival_summary)
```

```
## # A tibble: 2 x 3
##   Survived Count Proportion
##      <int> <int>      <dbl>
## 1        0   815      0.623
## 2        1   494      0.377
```

```r
train_clean = train %>%
  select(-PassengerId, -Name, -Ticket, -Cabin) %>%
  mutate(Sex = factor(Sex),
         Embarked = factor(Embarked),
         Survived = factor(Survived, levels = c(0, 1)))

missing_summary = colSums(is.na(train_clean) | train_clean == "")
print("Valores faltantes por columna:")
```

```
## [1] "Valores faltantes por columna:"
```

```r
print(missing_summary)
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0      175        0        0        0        2
```

```r
train_clean = train_clean %>%
  drop_na()
```

```r
model_full = glm(Survived ~ ., data = train_clean, family = binomial)
summary(model_full)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = train_clean)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.224161   0.670456   7.792 6.60e-15 ***
## Pclass      -1.021788   0.175166  -5.833 5.44e-09 ***
## Sexmale     -3.441495   0.241806 -14.232  < 2e-16 ***
## Age         -0.036108   0.008750  -4.127 3.68e-05 ***
## SibSp       -0.386168   0.140889  -2.741  0.00613 **
## Parch        0.035687   0.150111   0.238  0.81208
## Fare         0.002317   0.002385   0.971  0.33133
## EmbarkedQ   -0.167465   0.613427  -0.273  0.78485
## EmbarkedS   -0.214048   0.295694  -0.724  0.46914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 992.56  on 738  degrees of freedom
## Residual deviance: 574.94  on 730  degrees of freedom
## AIC: 592.94
##
```

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

```r
model_1 = glm(Survived ~ Pclass + Sex + Age,
              data = train_clean, family = binomial)
summary(model_1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age, family = binomial,
##     data = train_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.99484    0.53603   9.318  < 2e-16 ***
## Pclass      -1.12058    0.14633  -7.658 1.89e-14 ***
## Sexmale     -3.35814    0.22683 -14.805  < 2e-16 ***
## Age         -0.03184    0.00822  -3.873 0.000107 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 992.56  on 738  degrees of freedom
## Residual deviance: 585.20  on 735  degrees of freedom
## AIC: 593.2
##
## Number of Fisher Scoring iterations: 5
```

```r
model_2 = glm(Survived ~ Pclass + Sex + Age + Fare + Embarked,
              data = train_clean, family = binomial)
summary(model_2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + Fare + Embarked,
##     family = binomial, data = train_clean)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.918592   0.643916   7.639 2.20e-14 ***
## Pclass      -1.024670   0.172923  -5.926 3.11e-09 ***
## Sexmale     -3.338247   0.227818 -14.653  < 2e-16 ***
## Age         -0.030757   0.008325  -3.695  0.00022 ***
## Fare         0.001278   0.002191   0.583  0.55979
## EmbarkedQ   -0.114428   0.589370  -0.194  0.84606
## EmbarkedS   -0.284243   0.291687  -0.974  0.32982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 992.56  on 738  degrees of freedom
## Residual deviance: 583.42  on 732  degrees of freedom
## AIC: 597.42
##
```

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

```r
aic_comparison = data.frame(
  Modelo = c("Completo", "Modelo 1", "Modelo 2"),
  AIC = c(AIC(model_full), AIC(model_1), AIC(model_2))
)

print(aic_comparison)
```

```
##     Modelo      AIC
## 1 Completo 592.9429
## 2 Modelo 1 593.2001
## 3 Modelo 2 597.4162
```

```r
pred_1 = predict(model_1, newdata = train_clean, type = "response")
roc_1 = roc(train_clean$Survived, pred_1)
```

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

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

```r
auc_1 = auc(roc_1)
print(paste("AUC del modelo modelo 1:", auc_1))
```

```
## [1] "AUC del modelo modelo 1: 0.885060989608044"
```

```r
pred_2 = predict(model_2, newdata = train_clean, type = "response")
roc_2 = roc(train_clean$Survived, pred_2)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
auc_2 = auc(roc_2)
print(paste("AUC del modelo 2:", auc_2))
```

```
## [1] "AUC del modelo 2: 0.886706255069713"
```

```r
null_deviance_modelo1 = deviance(glm(Survived ~ 1, data = train_clean, family = binomial))
null_deviance_modelo2 = deviance(glm(Survived ~ 1, data = train_clean, family = binomial))

residual_deviance_modelo1 = deviance(model_1)
residual_deviance_modelo2 = deviance(model_2)

explained_deviance_modelo1 = null_deviance_modelo1 - residual_deviance_modelo1
explained_deviance_modelo2 = null_deviance_modelo2 - residual_deviance_modelo2

deviance_explained_comparison = data.frame(
  Modelo = c("Modelo 1", "Modelo 2"),
  Desviación_Nula = c(null_deviance_modelo1, null_deviance_modelo2),
  Desviación_Residual = c(residual_deviance_modelo1, residual_deviance_modelo2),
  Desviación_Explicada = c(explained_deviance_modelo1, explained_deviance_modelo2)
)

print(deviance_explained_comparison)
```

```
##     Modelo Desviación_Nula Desviación_Residual Desviación_Explicada
## 1 Modelo 1        992.5647            585.2001             407.3646
## 2 Modelo 2        992.5647            583.4162             409.1485
```

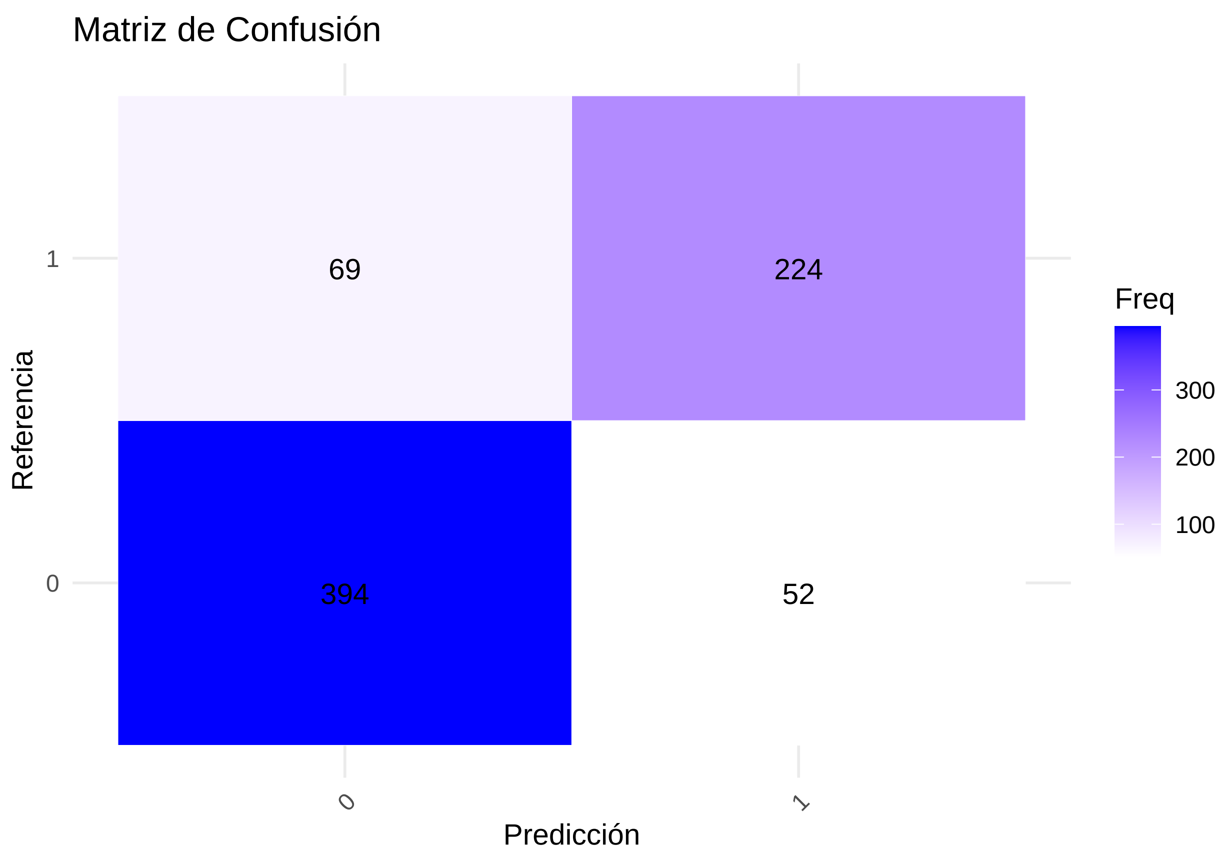
Mejor Modelo

El modelo 2 es más complejo, tiene un mejor ajuste y mejor capacidad de clasificación, lo que lo convierte en el mejor modelo en este caso. La diferencia en las métricas no es enorme, pero el Modelo 2 tiene una ligera ventaja en todos los aspectos clave. Tambien sugiere que la clase social, el sexo, la tarifa pagada, y el puerto de embarque tienen un impacto significativo en la probabilidad de supervivencia en el Titanic. El sexo femenino tiene un efecto positivo significativo en la supervivencia, mientras que la clase baja y la edad avanzada tienen un efecto negativo.

```r
pred_prob = predict(model_2, newdata = train_clean, type = "response")

pred_class = ifelse(pred_prob > 0.5, 1, 0)

conf_matrix = confusionMatrix(factor(pred_class), factor(train_clean$Survived))

conf_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 394  69
##          1  52 224
##
##                Accuracy : 0.8363
##                  95% CI : (0.8076, 0.8622)
##     No Information Rate : 0.6035
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6544
##
##  Mcnemar's Test P-Value : 0.1458
##
##             Sensitivity : 0.8834
##             Specificity : 0.7645
##          Pos Pred Value : 0.8510
##          Neg Pred Value : 0.8116
##              Prevalence : 0.6035
##          Detection Rate : 0.5332
##    Detection Prevalence : 0.6265
##       Balanced Accuracy : 0.8240
##
##        'Positive' Class : 0
##
```

```r
conf_matrix_data = as.data.frame(conf_matrix$table)

ggplot(conf_matrix_data, aes(x = Prediction, y = Reference)) +
  geom_tile(aes(fill = Freq), color = "white") +
  scale_fill_gradient(low = "white", high = "blue") +
  geom_text(aes(label = Freq), vjust = 1) +
  theme_minimal() +
  labs(title = "Matriz de Confusión", x = "Predicción", y = "Referencia") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

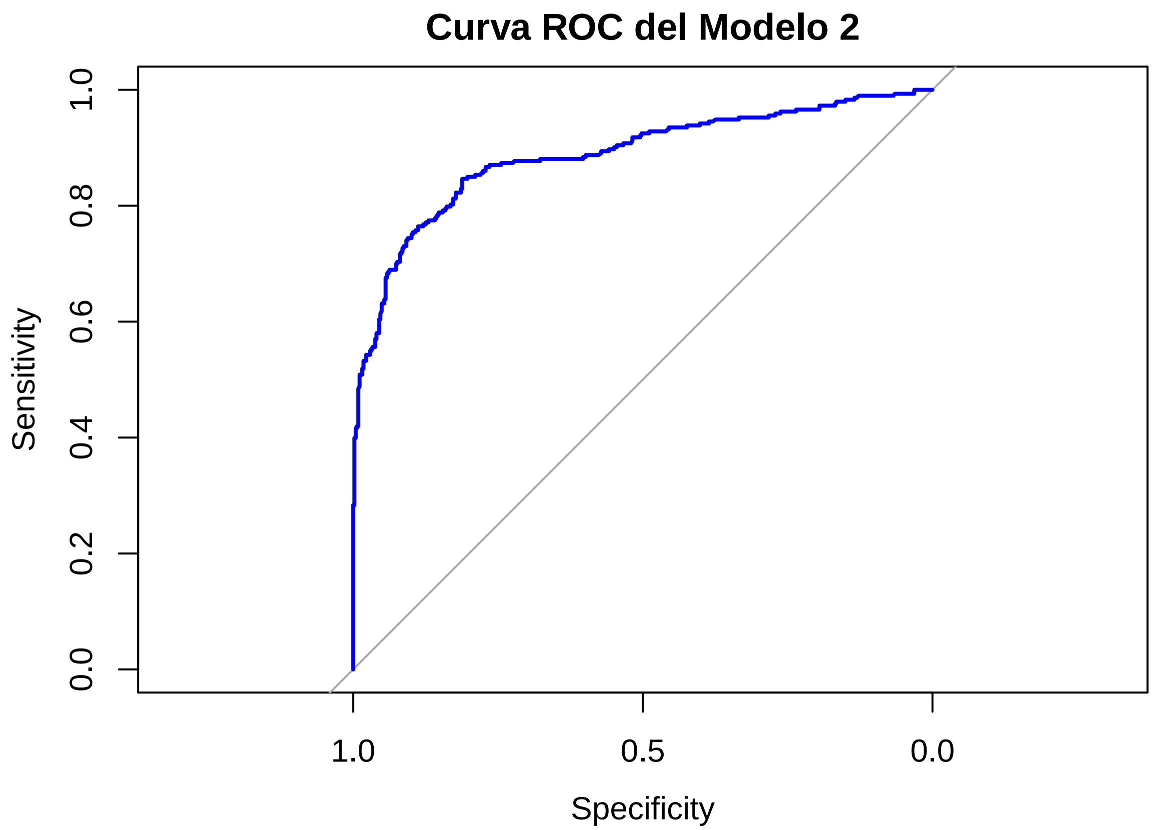



```r
roc_curve = roc(train_clean$Survived, pred_prob)
```

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

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

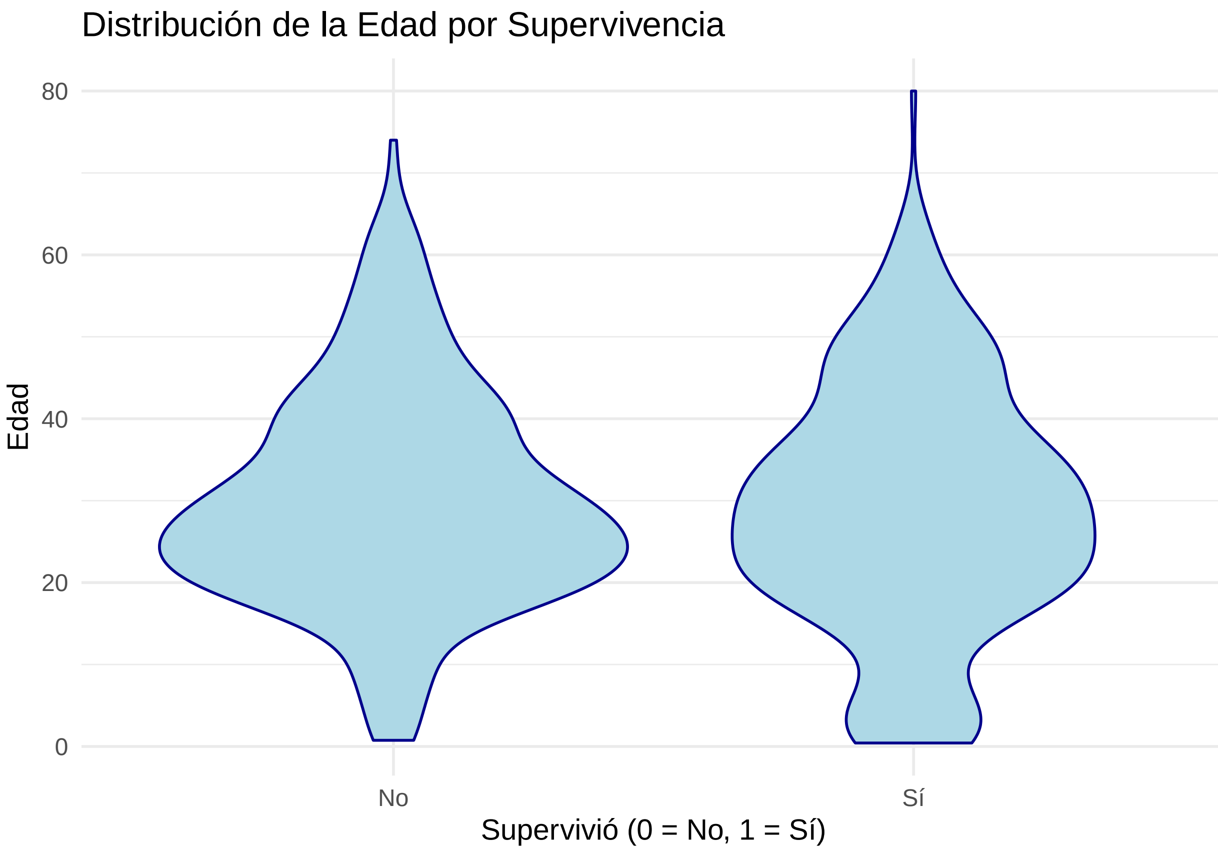
```r
plot(roc_curve, main = "Curva ROC del Modelo 2", col = "blue", lwd = 2)
```



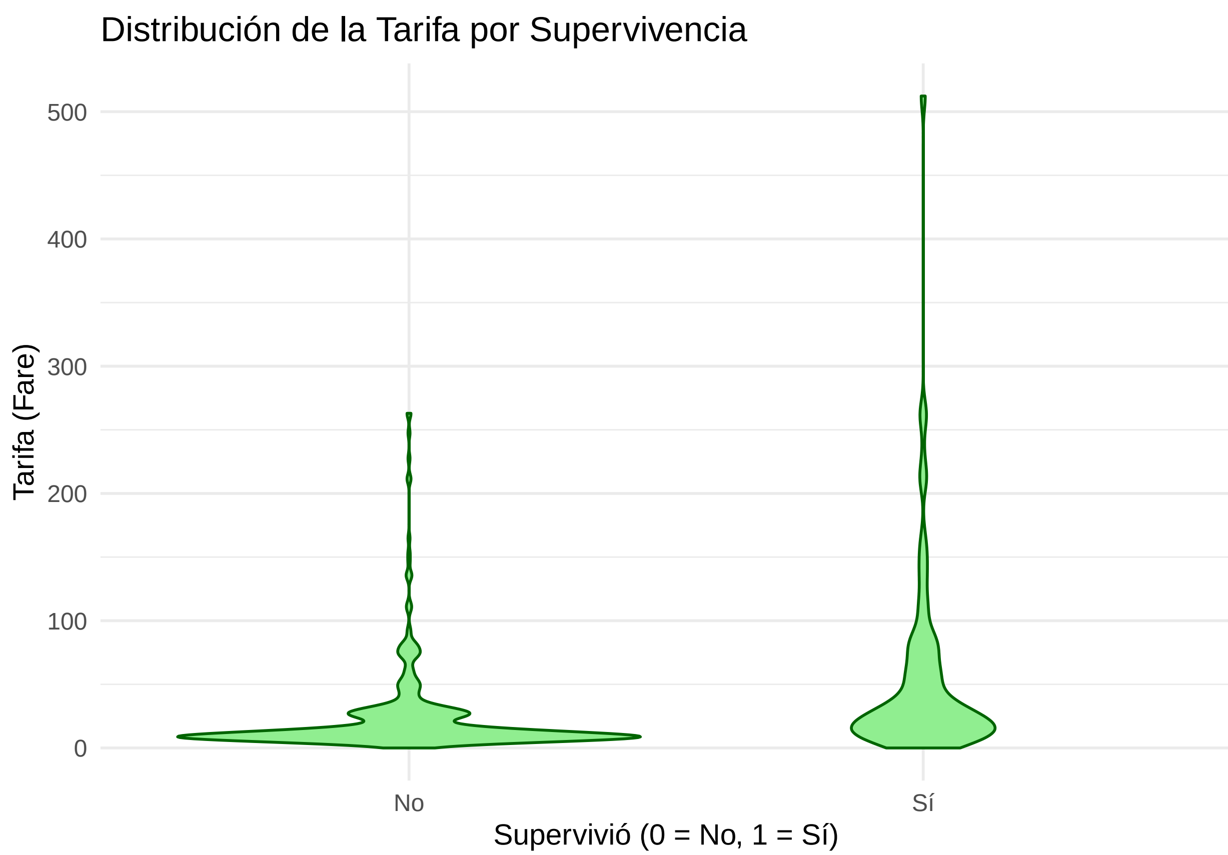

```
auc_value = auc(roc_curve)

print(paste("AUC del Modelo 2:", auc_value))
```

```
## [1] "AUC del Modelo 2: 0.886706255069713"
```

```
ggplot(train_clean, aes(x = factor(Survived), y = Age)) +
  geom_violin(fill = "lightblue", color = "darkblue") +
  labs(title = "Distribución de la Edad por Supervivencia",
       x = "Supervivió (0 = No, 1 = Sí)",
       y = "Edad") +
  theme_minimal() +
  scale_x_discrete(labels = c("No", "Sí"))
```




```r
ggplot(train_clean, aes(x = factor(Survived), y = Fare)) +
  geom_violin(fill = "lightgreen", color = "darkgreen") +
  labs(title = "Distribución de la Tarifa por Supervivencia",
       x = "Supervivió (0 = No, 1 = Sí)",
       y = "Tarifa (Fare)") +
  theme_minimal() +
  scale_x_discrete(labels = c("No", "Sí"))
```



Conclusion del Modelo

El Modelo 2 tiene un buen rendimiento general con una precisión del 83.63%, un AUC de 0.8867. El modelo tiene una buena capacidad predictiva, y podría beneficiarse de ajustes adicionales o pruebas con más variables para mejorar más la clasificación de los no sobrevivientes.

```r
validation = read.csv("documents/Titanic_test.csv")

validation_clean = validation %>%
  mutate(Sex = factor(Sex, levels = c("male", "female")),
         Embarked = factor(Embarked, levels = c("C", "Q", "S"))) %>%
  mutate(Age = ifelse(is.na(Age), median(Age, na.rm = TRUE), Age),
         Fare = ifelse(is.na(Fare), median(Fare, na.rm = TRUE), Fare))

prob_pred = predict(model_2, newdata = validation_clean, type = "response")

predicciones = ifelse(prob_pred > 0.5, 1, 0)

predicciones_df = data.frame(PassengerId = validation_clean$PassengerId, Survived = predicciones)

head(predicciones_df)
```

```
##   PassengerId Survived
## 1         892        0
## 2         893        1
## 3         894        0
## 4         895        0
```

```
## 5         896        1
## 6         897        0
```

```r
threshold_optimal <- 0.514

predicciones_optimas <- ifelse(prob_pred > threshold_optimal, 1, 0)

matriz_confusion_train <- confusionMatrix(factor(predicciones_optimas), factor(predicciones_df$Survived)

print(matriz_confusion_train)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 264   0
##          1   0 154
##
##                Accuracy : 1
##                  95% CI : (0.9912, 1)
##     No Information Rate : 0.6316
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6316
##          Detection Rate : 0.6316
##    Detection Prevalence : 0.6316
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

```r
matriz_confusion_train$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   1.000000e+00   1.000000e+00   9.912138e-01   1.000000e+00   6.315789e-01
## AccuracyPValue  McnemarPValue
##   3.791016e-84            NaN
```

```r
matriz_confusion_train$byClass
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            1.0000000            1.0000000            1.0000000
##       Neg Pred Value            Precision               Recall
##            1.0000000            1.0000000            1.0000000
##                   F1           Prevalence       Detection Rate
##            1.0000000            0.6315789            0.6315789
## Detection Prevalence    Balanced Accuracy
##            0.6315789            1.0000000
```

```r
titanic_test_clean = titanic_test %>%
  mutate(Sex = factor(Sex, levels = c("male", "female")),
         Embarked = factor(Embarked, levels = c("C", "Q", "S")),
         Age = ifelse(is.na(Age), median(train_clean$Age, na.rm = TRUE), Age),
         Fare = ifelse(is.na(Fare), median(train_clean$Fare, na.rm = TRUE), Fare))

prob_pred_test = predict(model_2, newdata = titanic_test_clean, type = "response")

threshold_optimal = 0.5
predictions_test = ifelse(prob_pred_test > threshold_optimal, 1, 0)

output_test = data.frame(PassengerId = titanic_test_clean$PassengerId,
                         Survived = predictions_test)

write.csv(output_test, "Predicciones_Titanic_Test.csv", row.names = FALSE)

head(output_test)
```

```
##   PassengerId Survived
## 1         892        0
## 2         893        1
## 3         894        0
## 4         895        0
## 5         896        1
## 6         897        0
```

Conclusion

Es importante tener en cuenta que este tipo de resultados en los datos de entrenamiento pueden ser una señal de sobreajuste. Esto significa que el modelo podría estar ajustándose demasiado a los datos específicos de este conjunto, y no necesariamente será capaz de predecir con la misma precisión en nuevos datos o en el conjunto de validación.

Tambien debido a que hay más pasajeros que no sobrevivieron (clase 0), el modelo podría estar favoreciendo esta clase, lo que a veces puede generar una alta precisión sin realmente estar haciendo predicciones muy útiles para la clase menos frecuente