

EXAMEN 4: MODELOS DE CLASIFICACIÓN

Bonilla Flores Emmanuel

García Morales Marcos

Hernández Galicia Alberto

Leyva Díaz Eduardo Tomás

2025-06-12

LIBRERÍAS

```
library(readxl)
library(MASS) #Para poder usar la función Lda/qda
library(nortest)
library(ggpubr)
library(MVN)
library(biotools) # test de Matriz de Covarianzas
library(ggplot2)
library(caTools)
library(caret)
library(purrr) # La función map
library(patchwork) # Ordenar gráficos ggplot2
library(GGally) #pairplot
library(tidyr) # Para utilizar la función drop.na
library(dplyr)
library(car)
library(rpart)
library(rpart.plot)
library(randomForest) # Librería específica para el modelo de Random Forest
library(MLmetrics) # Para F1_Score
setwd("~/Octavo Semestre/Análisis de Datos/Machine Learning")
```

PREPARACIÓN DE DATOS

Base de Datos

Este conjunto de datos contiene 11 características médicas que pueden utilizarse para predecir una posible enfermedad cardíaca, fue obtenido del repositorio en línea *Kaggle*, una plataforma de ciencia de datos que permite compartir y acceder a datasets públicos para el desarrollo de modelos de aprendizaje automático y análisis estadístico.

-Soriano, F. (2021). Heart Failure Prediction Dataset [Dataset]. Kaggle. <https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction> (<https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>)

Para la construcción del dataset se integraron datos clínicos de cinco bases públicas (Statlog, Cleveland, Hungarian, Switzerland y Long Beach VA), cada una con registros similares en las 11 columnas principales generando 918 observaciones (filas) y 12 variables (columnas) considerando la predicción. En la siguiente tabla se puede ver un fragmento de estos datos:

```
heart<-read.csv("heart.csv")
head(heart[,1:6],10)
```

##	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS
## 1	40	M	ATA	140	289	0
## 2	49	F	NAP	160	180	0
## 3	37	M	ATA	130	283	0
## 4	48	F	ASY	138	214	0
## 5	54	M	NAP	150	195	0
## 6	39	M	NAP	120	339	0
## 7	45	F	ATA	130	237	0
## 8	54	M	ATA	110	208	0
## 9	37	M	ASY	140	207	0
## 10	48	F	ATA	120	284	0

```
head(heart[,7:12],10)
```

```
##      RestingECG MaxHR ExerciseAngina Oldpeak ST_Slope HeartDisease
## 1      Normal   172           N      0.0      Up           0
## 2      Normal   156           N      1.0      Flat          1
## 3         ST    98           N      0.0      Up           0
## 4      Normal   108           Y      1.5      Flat          1
## 5      Normal   122           N      0.0      Up           0
## 6      Normal   170           N      0.0      Up           0
## 7      Normal   170           N      0.0      Up           0
## 8      Normal   142           N      0.0      Up           0
## 9      Normal   130           Y      1.5      Flat          1
## 10     Normal   120           N      0.0      Up           0
```

Descripción de Variables

De las 12 variables de todo el conjunto de datos hay algunas que son numéricas y otras que son categóricas, a continuación se presenta una breve descripción de cada una para entender de mejor manera los valores disponibles.

NUMÉRICAS

- **Age:** Edad del paciente (años).
- **RestingBP:** Presión arterial en reposo (mm Hg).
- **Cholesterol:** Nivel de colesterol (mg/dl).
- **MaxHR:** Frecuencia cardíaca máxima alcanzada durante una prueba de esfuerzo.
- **Oldpeak:** Depresión del segmento ST inducida por ejercicio.

CATEGÓRICAS

- **Sex:** Sexo del paciente (M: Male, F: Female).
- **ChestPainType:** Tipo de dolor en el pecho, hay cuatro resultados posibles: TA-Angina Típica, ATA-Angina Atípica, NAP-Dolor no anginoso y ASY-Asintomático.
- **FastingBS:** Glucosa en ayunas, si es mayor a 120 mg/dl se le asigna el valor de 1 (hiperglucemia) y si no toma el valor de 0.
- **RestingECG:** Electrocardiograma en reposo, puede resultar en Normal, LVH y ST.
- **ExerciseAngina:** Angina inducida por ejercicio, si hay un dolor o molestia en el pecho cuando la persona hace actividad física vale 1 y si no 0.
- **ST_Slope:** Pendiente del segmento ST al final del ejercicio, puede ser Up (ascendente), Down (descendente) o Flat (plana).
- **HeartDisease:** Es el target o la clase objetivo, indica la presencia (1) o ausencia (0) de la enfermedad cardíaca.

Al momento de realizar los modelos se tomarán en cuenta todas las variables debido a que representan distintos aspectos de la salud física y hábitos de vida que combinados permiten identificar señales tempranas o patrones comunes en personas con enfermedad cardíaca, es decir, ayuda a médicos o investigadores a determinar factores de riesgo clave.

Propósito

Se desarrollarán y evaluarán cuatro modelos de clasificación supervisada a través de una Matriz de Confusión donde se permite visualizar los aciertos o errores de cada modelo (métricas), estos son:

- Regresión Logística
- Análisis Discriminante Cuadrático (QDA)
- Árbol de Decisión
- Bosque Aleatorio

El objetivo principal es seleccionar el modelo que logre una clasificación precisa entre personas con y sin enfermedad cardíaca, priorizando la capacidad del modelo para identificar correctamente a los pacientes que sí presentan la enfermedad. Por esta razón se dará especial énfasis a la **Sensibilidad o Recall**, métrica clave para minimizar los errores Tipo II (falsos negativos), ya que estos representan un riesgo mayor al no detectar a personas que requieren atención médica.

Limpieza y Preparación

Antes de realizar cualquier cosa se va a generar una copia de la base de datos para que la información original no sufra cambios. Es en ese nuevo data frame donde se harán todas las modificaciones comenzando por el nombre de las columnas para hacer referencia con mayor facilidad y también para la interpretación de resultados.

```
# Copia
enfermedad<-heart
colnames(enfermedad)<-c("Edad", "Sexo", "Dolor_Pecho", "Presión_Reposo", "Colesterol",
                        "Glucosa_Ayunas", "Electrocardiograma_Reposo", "Frecuencia_Máxima",
                        "Angina_Ejercicio", "Depresión_ST", "Pendiente_ST", "Enfermedad_Cardíaca")
head(enfermedad)
```

```
##   Edad Sexo Dolor_Pecho Presión_Reposo Colesterol Glucosa_Ayunas
## 1   40   M      ATA      140      289      0
## 2   49   F      NAP      160      180      0
## 3   37   M      ATA      130      283      0
## 4   48   F      ASY      138      214      0
## 5   54   M      NAP      150      195      0
## 6   39   M      NAP      120      339      0
##   Electrocardiograma_Reposo Frecuencia_Máxima Angina_Ejercicio Depresión_ST
## 1                      Normal      172      N      0.0
## 2                      Normal      156      N      1.0
## 3                      ST      98      N      0.0
## 4                      Normal      108      Y      1.5
## 5                      Normal      122      N      0.0
## 6                      Normal      170      N      0.0
##   Pendiente_ST Enfermedad_Cardíaca
## 1          Up      0
## 2         Flat      1
## 3          Up      0
## 4         Flat      1
## 5          Up      0
## 6          Up      0
```

Valores Faltantes

Verificamos si tiene registros que estén incompletos para tomar una decisión de que hacer con ellos antes de hacer cualquier análisis.

```
colSums(is.na(enfermedad))
```

```
##           Edad           Sexo           Dolor_Pecho
##           0             0             0
##   Presión_Reposo      Colesterol      Glucosa_Ayunas
##           0             0             0
## Electrocardiograma_Reposo Frecuencia_Máxima Angina_Ejercicio
##           0             0             0
##           Depresión_ST      Pendiente_ST Enfermedad_Cardíaca
##           0             0             0
```

La información está completa y no le falta ningún dato.

Variables Categóricas

Como ya se mencionó, dentro de la base de datos tenemos algunas variables que son binarias como el Sexo, Glucosa_Ayunas, o Angina_Ejercicio, además de otras que tienen más valores posibles como Dolor_Pecho, Electrocardiograma_Reposo o Pendiente_SP. Lo que se va a hacer es convertirlas a factor para que puedan ser consideradas en el modelo.

NOTA: En algunos casos se cambiaron los valores originales por otros que son más fáciles de entender y de interpretar.

```
# Convertirlos a Factor
enfermedad$Enfermedad_Cardíaca<-factor(enfermedad$Enfermedad_Cardíaca,
                                       levels=c(0,1),labels=c("Sin Enfermedad", "Con Enfermedad"))
enfermedad$Dolor_Pecho<-factor(enfermedad$Dolor_Pecho)
enfermedad$Glucosa_Ayunas<-factor(enfermedad$Glucosa_Ayunas,levels=c(0,1),labels=c("No", "Sí"))
enfermedad$Electrocardiograma_Reposo<-factor(enfermedad$Electrocardiograma_Reposo)
enfermedad$Angina_Ejercicio<-factor(enfermedad$Angina_Ejercicio,
                                    levels=c("N", "Y"),labels=c("No", "Sí"))
enfermedad$Pendiente_ST<-factor(enfermedad$Pendiente_ST)

glimpse(enfermedad)
```

```
## Rows: 918
## Columns: 12
## $ Edad          <int> 40, 49, 37, 48, 54, 39, 45, 54, 37, 48, 37, ...
## $ Sexo          <chr> "M", "F", "M", "F", "M", "M", "F", "M", "M", ...
## $ Dolor_Pecho   <fct> ATA, NAP, ATA, ASY, NAP, NAP, ATA, ATA, ASY, ...
## $ Presión_Reposo <int> 140, 160, 130, 138, 150, 120, 130, 110, 140, ...
## $ Colesterol    <int> 289, 180, 283, 214, 195, 339, 237, 208, 207, ...
## $ Glucosa_Ayunas <fct> No, No, No, No, No, No, No, No, No, No, No, ...
## $ Electrocardiograma_Reposo <fct> Normal, Normal, ST, Normal, Normal, Normal, ...
## $ Frecuencia_Máxima <int> 172, 156, 98, 108, 122, 170, 170, 142, 130, ...
## $ Angina_Ejercicio <fct> No, No, No, Sí, No, No, No, No, Sí, No, No, ...
## $ Depresión_ST    <dbl> 0.0, 1.0, 0.0, 1.5, 0.0, 0.0, 0.0, 0.0, 1.5, ...
## $ Pendiente_ST    <fct> Up, Flat, Up, Flat, Up, Up, Up, Up, Flat, Up, ...
## $ Enfermedad_Cardíaca <fct> Sin Enfermedad, Con Enfermedad, Sin Enfermedad, ...
```

Manipulación de Datos

Lo primero que se va a realizar es un análisis exploratorio de la base para conocer un poco más de información sobre las variables predictoras y de respuesta, específicamente se hará un resumen estadístico con las medidas más importantes acompañadas de gráficos como boxplots o histogramas.

Resumen Estadístico

Es posible armar una tabla que contenga la información estadística más relevante de las variables numéricas debido a que nos ayudará a saber el comportamiento general de los datos.

```
# Variables Numéricas
num_vars <- sapply(enfermedad, is.numeric)
enfermedad_num <- enfermedad[, num_vars]

resumen<-function(x){
  y<-enfermedad_num[[x]]      # vector con los datos
  df<-data.frame(Variable=x)
  df<-df %>% mutate(Mínimo=min(y),Bigote_Inferior=boxplot.stats(y)$stats[1],
                    Q1=quantile(y,0.25),Mediana = median(y), Media = mean(y),
                    Q3=quantile(y,0.75), Bigote_Superior=boxplot.stats(y)$stats[5],
                    Máximo=max(y))
  return(df)
}
resumen_estadistico<-map_dfr(colnames(enfermedad_num),resumen)
resumen_estadistico
```

##	Variable	Mínimo	Bigote_Inferior	Q1	Mediana	Media	Q3
## 1	Edad	28.0	28	47.00	54.0	53.5108932	60.0
## 2	Presión_Reposo	0.0	92	120.00	130.0	132.3965142	140.0
## 3	Colesterol	0.0	85	173.25	223.0	198.7995643	267.0
## 4	Frecuencia_Máxima	60.0	67	120.00	138.0	136.8093682	156.0
## 5	Depresión_ST	-2.6	-2	0.00	0.6	0.8873638	1.5
##	Bigote_Superior	Máximo					
## 1		77.0	77.0				
## 2		170.0	200.0				
## 3		407.0	603.0				
## 4		202.0	202.0				
## 5		3.7	6.2				

Una vez armada la tabla es posible rescatar lo siguiente:

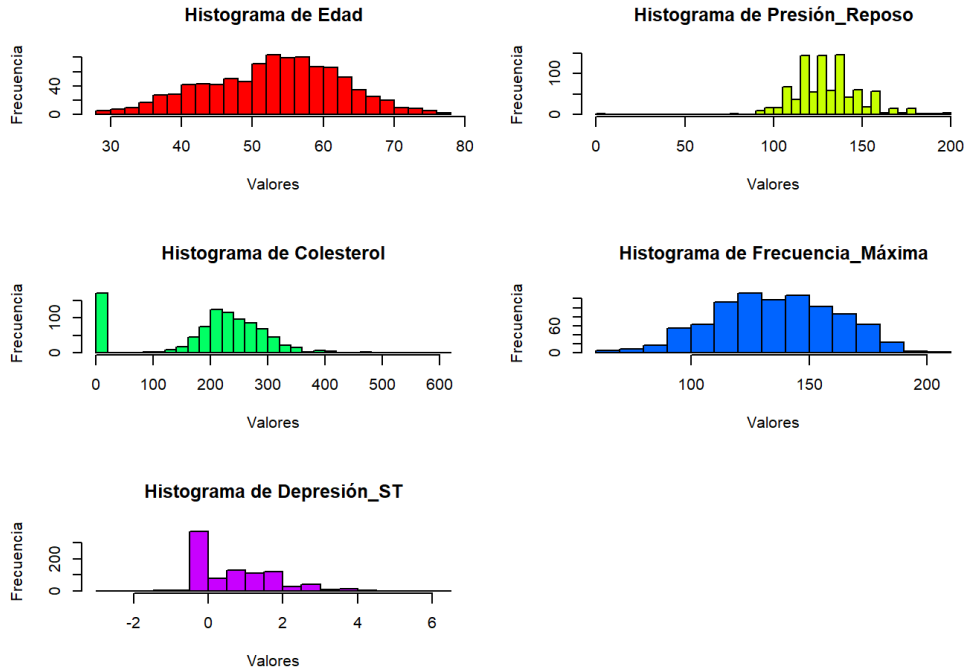
- La edad promedio es de 53.51 años, el paciente más joven tiene 28 y el más grande 77.
- El 25% de estos pacientes alcanzan una Frecuencia Máxima que supera los 156, esto indica que se aceleraron mucho al hacer el esfuerzo físico, incluso el valor más alto es de 202.
- Hay casos donde posiblemente haya outliers debido a que los valores del Bigote Inferior o Superior no coinciden con el Mínimo o el Máximo, esto ocurre en todas las variables menos para la Edad.
- Viendo la asimetría de los datos, existe un sesgo hacia la izquierda Edad, Colesterol y Frecuencia Máxima (la mediana supera a la media), mientras que hay un sesgo a la derecha para la Presión en Reposo y la Depresión del Segmento T.
- Para el Colesterol y la Presión en Reposo aparecen valores iguales a 0, esto es improbable y puede ser que haya un error en el registro.

- La depresión ST es la única variable que tiene algunos valores negativos, el 50% es menor o igual a los 0.6.

Histogramas

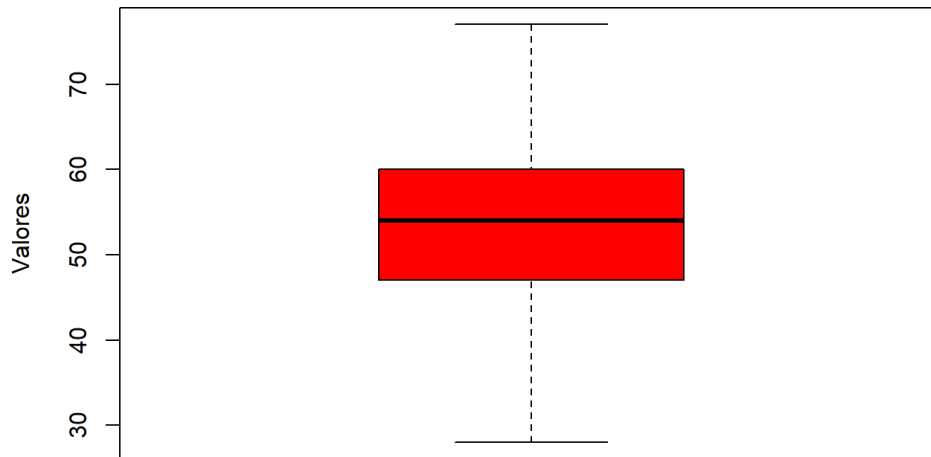
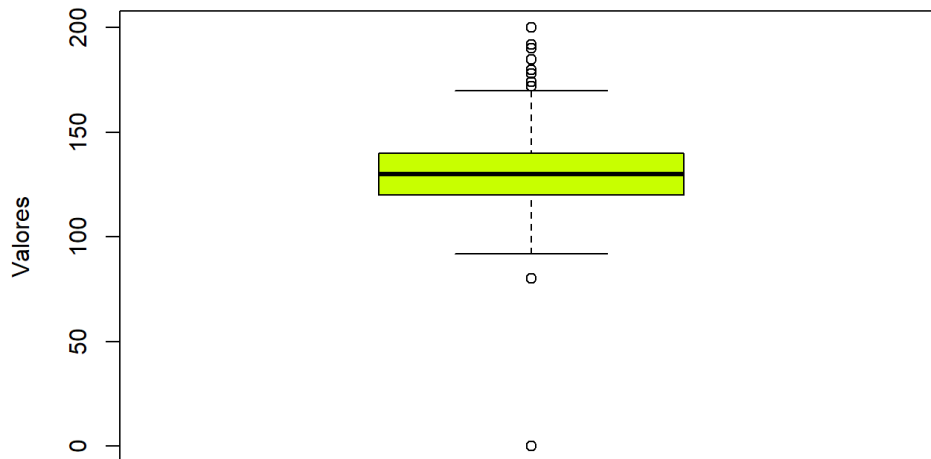
En los Histogramas de Frecuencia se puede apreciar los sesgos mencionados con anterioridad, por eso se le colocó el número de bins correspondiente a la regla de FD (datos con sesgos).

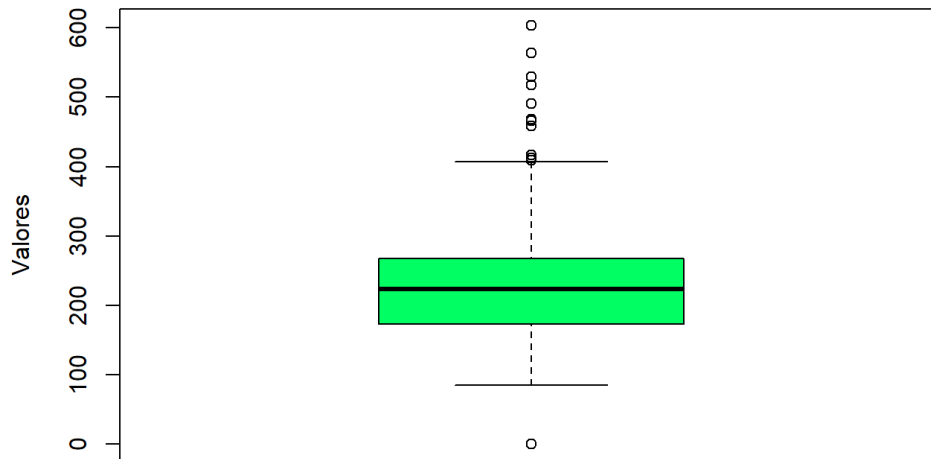
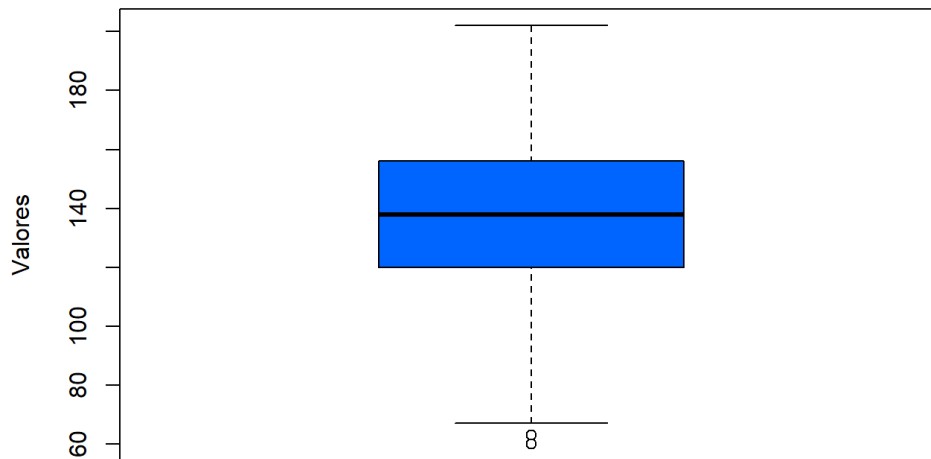
```
# HISTOGRAMAS
par(mfrow=c(3,2))
titulos<-colnames(enfermedad_num)
colores<-rainbow(length(titulos))
for(i in 1:length(titulos)){
  hist(enfermedad_num[[i]],col=colores[i],main=paste("Histograma de",titulos[i]),
    xlab="Valores",ylab="Frecuencia",breaks=nclass.FD(enfermedad_num[[i]]))
}
```



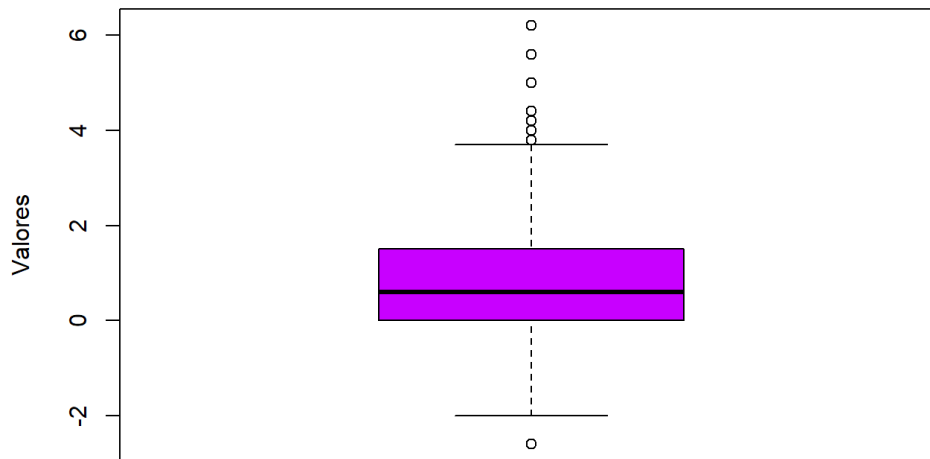
Boxplots

```
# BOXPLOTS
titulos<-colnames(enfermedad_num)
colores<-rainbow(length(titulos))
for(i in 1:length(titulos)){
  boxplot(enfermedad_num[[i]],col=colores[i],main=paste("Boxplot",titulos[i]),
    ylab="Valores")
}
```

Boxplot Edad**Boxplot Presión_Reposo**

Boxplot Colesterol**Boxplot Frecuencia_Máxima**

Boxplot Depresión_ST



- Aquí se confirma la presencia de los valores atípicos (puntos debajo o por arriba de los bigotes) en casi todas las variables.
- La Frecuencia Máxima y el Colesterol tienen el mayor rango intercuantil (IQR), lo cual muestra que para estas dos variables hay dispersión de los datos.
- Para la Presión en Reposo se ve que la caja es la más pequeña y equilibrada, además de que se observan los datos registrados como 0 cuando dentro del contexto es imposible porque los marca como outliers.

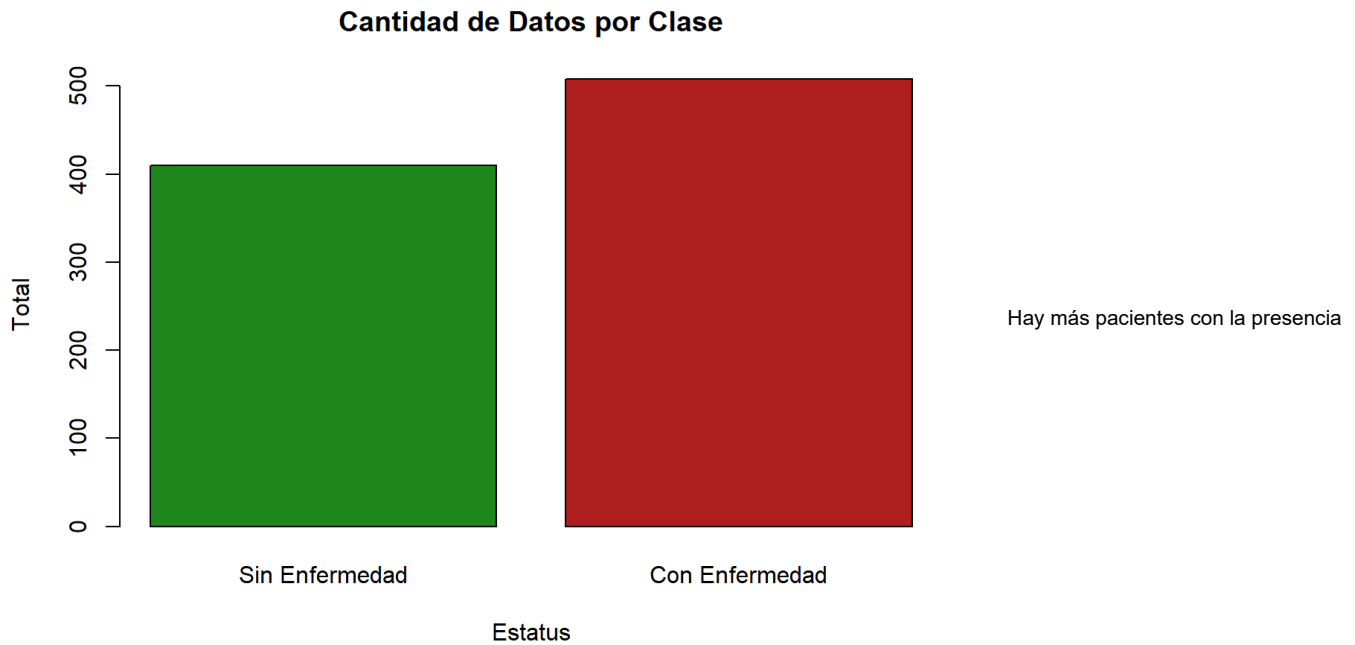
Proporciones

Es importante tener presente la cantidad de datos que corresponden a cada clase, esto para ver si no hay algún desbalance o que la mayoría de datos pertenezcan solamente a una porque puede impactar en las predicciones.

```
# Clases
proporciones<- enfermedad %>% group_by(Enfermedad_Cardiaca) %>% summarise(Total = n()) %>%
  mutate(Proporción=Total / sum(Total))
proporciones
```

```
## # A tibble: 2 × 3
##   Enfermedad_Cardiaca Total Proporción
##   <fct>              <int>      <dbl>
## 1 Sin Enfermedad      410      0.447
## 2 Con Enfermedad      508      0.553
```

```
barplot(Total ~ Enfermedad_Cardiaca, data = proporciones,xlab="Estatus",
  main="Cantidad de Datos por Clase",col=c("forestgreen","firebrick"))
```

de la enfermedad cardíaca, pero podemos ver que la proporción no varía mucho y están equilibrados.

Por otro lado, podemos observar los niveles o categorías de cada variable convertida a factor con el objetivo de tener presente cuáles son los posibles valores que pueden salir en cada una.

```
# Niveles
variables_factor <- sapply(enfermedad, is.factor)
for (variable in names(enfermedad)[variables_factor]) {
  cat("Variable:", variable, "\n")
  print(levels(enfermedad[[variable]]))
  cat("\n")
}
```

```
## Variable: Dolor_Pecho
## [1] "ASY" "ATA" "NAP" "TA"
##
## Variable: Glucosa_Ayunas
## [1] "No" "Sí"
##
## Variable: Electrocardiograma_Reposo
## [1] "LVH" "Normal" "ST"
##
## Variable: Angina_Ejercicio
## [1] "No" "Sí"
##
## Variable: Pendiente_ST
## [1] "Down" "Flat" "Up"
##
## Variable: Enfermedad_Cardiaca
## [1] "Sin Enfermedad" "Con Enfermedad"
```

```
head(heart[,c(1,4,5,8,10)],10)
```

##	Age	RestingBP	Cholesterol	MaxHR	Oldpeak
## 1	40	140	289	172	0.0
## 2	49	160	180	156	1.0
## 3	37	130	283	98	0.0
## 4	48	138	214	108	1.5
## 5	54	150	195	122	0.0
## 6	39	120	339	170	0.0
## 7	45	130	237	170	0.0
## 8	54	110	208	142	0.0
## 9	37	140	207	130	1.5
## 10	48	120	284	120	0.0

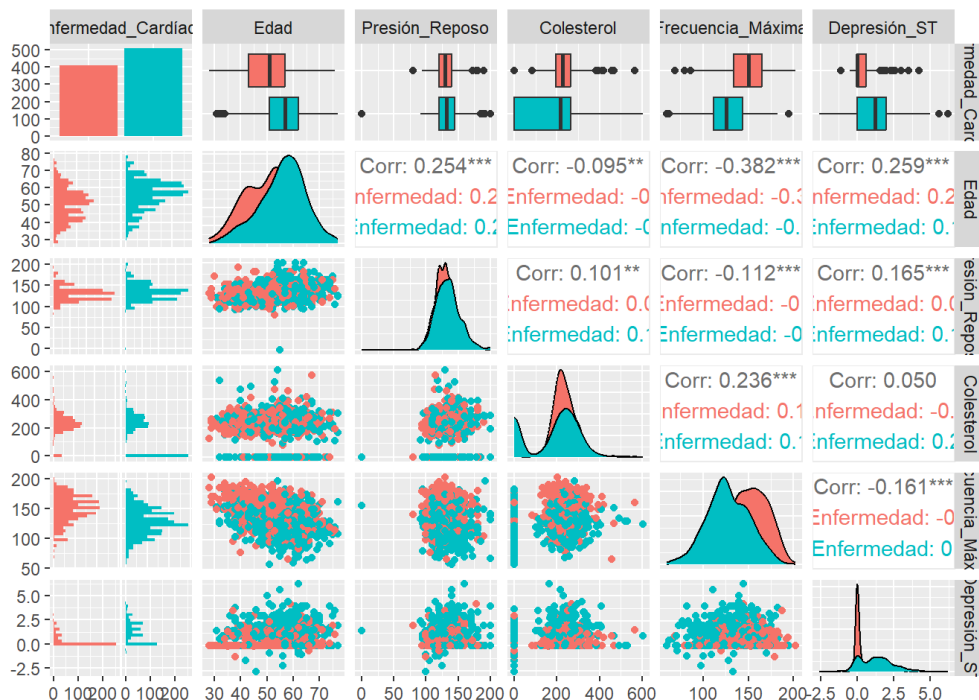
```
head(heart[, -c(1,4,5,8,10)],10)
```

##	Sex	ChestPainType	FastingBS	RestingECG	ExerciseAngina	ST_Slope	HeartDisease
## 1	M	ATA	0	Normal	N	Up	0
## 2	F	NAP	0	Normal	N	Flat	1
## 3	M	ATA	0	ST	N	Up	0
## 4	F	ASY	0	Normal	Y	Flat	1
## 5	M	NAP	0	Normal	N	Up	0
## 6	M	NAP	0	Normal	N	Up	0
## 7	F	ATA	0	Normal	N	Up	0
## 8	M	ATA	0	Normal	N	Up	0
## 9	M	ASY	0	Normal	Y	Flat	1
## 10	F	ATA	0	Normal	N	Up	0

MODELO LOGÍSTICO

Separación de Clases

```
# Hacer el ggpairs solo con las numéricas
enfermedad_num$Enfermedad_Cardiaca<-enfermedad$Enfermedad_Cardiaca
enfermedad_num<- enfermedad_num %>% dplyr::select(Enfermedad_Cardiaca,everything())
ggpairs(data = enfermedad_num, aes(color=Enfermedad_Cardiaca))
```



Boxplots

Comparan las distribuciones de variables continuas según la presencia o ausencia de enfermedad, estas mismas muestran la mediana, cuartiles y posibles outliers.

- Las personas con enfermedad cardíaca tienden a ser mayores que las que no la tienen.
- Las personas con enfermedad tienen, en general, menor frecuencia cardíaca máxima.
- En todas las variables existen valores atípicos, lo cual se debe tomar en cuenta para pruebas posteriores.

Histogramas

Representan la distribución de cada variable individualmente. Están separados por color según presencia o ausencia de enfermedad.

- No se observan diferencias claras entre los grupos, si acaso en la edad o en la Frecuencia Máxima.
- Para el histograma de Depresión ST se aprecia un pico en los valores para las personas que no tienen la enfermedad.

Graficos de dispersión

Muestran relaciones bivariadas entre pares de variables

Correlaciones Incluyen el coeficiente de Pearson, en este caso vemos el grado de correlación:

- Correlaciones bajas, ya sean positivas o negativas
- Edad correlaciona positivamente con Depresión_ST.

Aplicación del Modelo

Dividimos nuestros datos en “Entrenamiento” y “Prueba” aplicando una proporción de 80% y 20% respectivamente.

```
set.seed(123)
barajeado <- slice_sample(enfermedad, prop = 1)
split <- sample.split(barajeado$Edad, SplitRatio = 0.8) # Vector Lógico
enfermedad_train <- subset(barajeado, split == TRUE)
enfermedad_test <- subset(barajeado, split == FALSE)
head(enfermedad_train)
```

```
##      Edad Sexo Dolor_Pecho Presión_Reposo Colesterol Glucosa_Ayunas
## 3      37   M      NAP           130         194         No
## 5      41   F      ATA           125         184         No
## 6      60   M      ASY           125         258         No
## 8      51   M      ASY           110          0         Sí
## 10     43   F      ATA           120         266         No
## 11     49   M      ASY           140         234         No
##      Electrocardiograma_Reposo Frecuencia_Máxima Angina_Ejercicio Depresión_ST
## 3                        Normal           150         No           0.0
## 5                        Normal           180         No           0.0
## 6                        LVH           141         Sí           2.8
## 8                        Normal           92         No           0.0
## 10                       Normal          118         No           0.0
## 11                       Normal          140         Sí           1.0
##      Pendiente_ST Enfermedad_Cardíaca
## 3              Up      Sin Enfermedad
## 5              Up      Sin Enfermedad
## 6             Flat      Con Enfermedad
## 8             Flat      Con Enfermedad
## 10             Up      Sin Enfermedad
## 11             Flat      Con Enfermedad
```

```
head(enfermedad_test)
```

```
##      Edad Sexo Dolor_Pecho Presión_Reposo Colesterol Glucosa_Ayunas
## 1      54   M      ASY           130           0           Sí
## 2      59   M      ASY           122          233           No
## 4      45   M      NAP           130          236           No
## 7      59   F      ASY           130          338           Sí
## 9      41   M      ATA           120          295           No
## 25     60   M      ATA           160          267           Sí
##      Electrocardiograma_Reposo Frecuencia_Máxima Angina_Ejercicio Depresión_ST
## 1              Normal           110              Sí           3.0
## 2              Normal           117              Sí           1.3
## 4              Normal           144              No           0.1
## 7              ST             130              Sí           1.5
## 9              Normal           170              No           0.0
## 25             ST             157              No           0.5
##      Pendiente_ST Enfermedad_Cardíaca
## 1      Flat      Con Enfermedad
## 2      Down      Con Enfermedad
## 4      Up        Sin Enfermedad
## 7      Flat      Con Enfermedad
## 9      Up        Sin Enfermedad
## 25     Flat      Con Enfermedad
```

```
# Ajuste del modelo de regresión Logística
modelo_logistico<- glm(Enfermedad_Cardíaca~ .,
  data = enfermedad_train,
  family = binomial)
```

```
# Resumen del modelo
summary(modelo_logistico)
```

```
##
## Call:
## glm(formula = Enfermedad_Cardíaca ~ ., family = binomial, data = enfermedad_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.933773   1.592334   0.586 0.557595
## Edad           0.014353   0.014419   0.995 0.319515
## SexoM          1.494420   0.306023   4.883 1.04e-06 ***
## Dolor_PechoATA -1.719950   0.364735  -4.716 2.41e-06 ***
## Dolor_PechoNAP -1.667207   0.296148  -5.630 1.81e-08 ***
## Dolor_PechoTA  -1.756828   0.492318  -3.568 0.000359 ***
## Presión_Reposo -0.001541   0.006707  -0.230 0.818317
## Colesterol     -0.003420   0.001233  -2.773 0.005555 **
## Glucosa_AyunasSí 1.013087   0.306824   3.302 0.000961 ***
## Electrocardiograma_ReposoNormal -0.404531   0.301704  -1.341 0.179978
## Electrocardiograma_ReposoST -0.584791   0.392611  -1.489 0.136358
## Frecuencia_Máxima -0.008342   0.005588  -1.493 0.135460
## Angina_EjercicioSí 0.846031   0.277896   3.044 0.002331 **
## Depresión_ST    0.343046   0.130885   2.621 0.008768 **
## Pendiente_STFlat 0.912305   0.522237   1.747 0.080652 .
## Pendiente_STUp  -1.518302   0.549238  -2.764 0.005703 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1008.78  on 735  degrees of freedom
## Residual deviance:  482.37  on 720  degrees of freedom
## AIC: 514.37
##
## Number of Fisher Scoring iterations: 5
```

Antes de pasar a la interpretación de cada uno de los coeficientes, vamos a analizar por qué tiene sentido clínico que ciertas variables hayan resultado **significativas** en el modelo, veamos:

- SexoM: Ser hombre aumenta significativamente el riesgo de enfermedad cardíaca, esto porque los hombres suelen tener mayor riesgo cardiovascular que las mujeres antes de la menopausia por diferencias hormonales. Esto era esperado

- Dolor_Pecho: El dolor torácico típicamente anginoso es el que más se asocia a enfermedad coronaria. Los atípicos o no anginosos, aunque pueden ser dolorosos, son menos específicos de enfermedad cardíaca.
- Glucosa: Tener glucosa alta incrementa significativamente la probabilidad de enfermedad.
- Colesterol: El coeficiente negativo indica que a mayor nivel de colesterol, menor es la probabilidad de enfermedad cardíaca, lo cual parece contradictorio clínicamente.
- Glucosa_Ayunas: Tener glucosa alta incrementa significativamente la probabilidad de enfermedad.
- Depresión: Cada unidad de depresión del ST incrementa el riesgo de enfermedad cardíaca. Coherente con la evidencia médica.

Interpretación de Coeficientes

En términos de Log-odds y odds.

- SexoM (1.4944)(4.46) : Ser hombre multiplica por 4.46 las odds de tener enfermedad cardíaca
- Dolor_Pecho (-1.7199,-1.6672,-1.7568)(0.18,0.19,0.17): Dado que es negativo el coeficiente, esto nos quiere decir que reduce en (1-odd) la odd.
- Colesterol(-0.0034)(0.9966): Cada unidad adicional de colesterol reduce las odds en 0.34%.
- Glucosa_AyunasSí (1.013087)(2.75): Si tiene glucosa en ayunas alta, las odds aumentan 2.75 veces.
- Angina_EjercicioSí (0.846031)(2.33) : Tener angina con ejercicio duplica las odds de enfermedad cardíaca.
- Depresión_ST (0.343046)(1.41) : Cada unidad más de depresión ST aumenta las odds en 41%.
- Pendiente_STUp (-1.518302)(0.22) : Reduce las odds en 78% si la pendiente del ST sube.

Matriz de Confusión

El siguiente paso es hacer la evaluación del modelo, así que se calculan las predicciones con el conjunto de prueba y se genera la matriz de confusión.

```
# Paso 1: Calcular las probabilidades para los datos de prueba
probabilidades <- predict(modelo_logistico,newdata=enfermedad_test,type="response")
head(probabilidades)
```

```
##           1           2           4           7           9           25
## 0.99585363 0.90040092 0.06354911 0.88505106 0.03776027 0.64845292
```

```
# Paso 2: Convertir probabilidades en clases 0 o 1 con umbral 0.5
predicciones_clase <- ifelse(probabilidades < 0.5,"Sin Enfermedad","Con Enfermedad")
head(predicciones_clase)
```

```
##           1           2           4           7
## "Con Enfermedad" "Con Enfermedad" "Sin Enfermedad" "Con Enfermedad"
##           9           25
## "Sin Enfermedad" "Con Enfermedad"
```

```
# Hay que asegurarse que ambas estén como factores
y_real<-enfermedad_test$Enfermedad_Cardíaca
y_pred<-factor(predicciones_clase,levels=c("Sin Enfermedad","Con Enfermedad"))

# Matriz de confusión y métricas
conf_matrix <- confusionMatrix(data=y_pred,reference=y_real,positive="Con Enfermedad")
print(conf_matrix$table)
```

```
##           Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad          71           8
## Con Enfermedad          17          86
```

```
# Métricas individuales
accuracy <- conf_matrix$overall["Accuracy"] # Proporción total de predicciones correctas.
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8626374
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.8349515
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.9148936
```

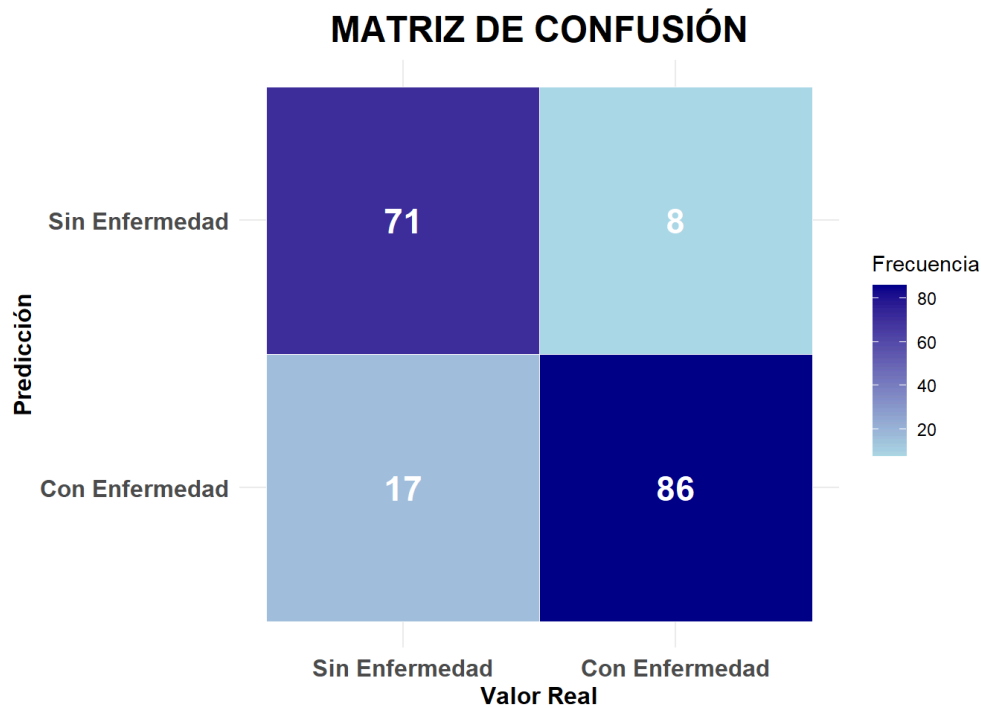
```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.8730964
```

```
cm_table <- as.data.frame(conf_matrix$table)

# Invertir el orden de los niveles del eje Y
cm_table$Prediction <- factor(cm_table$Prediction, levels = rev(levels(cm_table$Prediction)))

ggplot(cm_table, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), color = "white", size = 6, fontface = "bold") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name = "Frecuencia") +
  labs(
    title = "MATRIZ DE CONFUSIÓN",
    x = "Valor Real",
    y = "Predicción"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 12, face = "bold")
  )
```



- Accuracy(Exactitud): El modelo clasifica correctamente al 86.3% de los casos
- Precision (Precisión): De todas las personas clasificadas como con enfermedad, el 83.5% realmente la tienen.
- Recall (Sensibilidad): De todas las personas que realmente tienen enfermedad cardíaca, el 91.5% fueron correctamente detectadas.
- F1 Score: El modelo tiene un muy buen balance entre precisión y sensibilidad.El modelo tiene balance entre detectar correctamente a los enfermos (recall) y no generar demasiados falsos positivos (precisión).

En el contexto de salud, un falso negativo (no detectar a alguien enfermo) puede tener consecuencias graves. Por eso, la alta sensibilidad (91.5%) es la métrica más relevante en este caso. El modelo prioriza correctamente la detección de pacientes con riesgo, lo cual es deseable para la prevención y tratamiento oportuno.

Modelo Step

A continuación, utilizamos el método `step()` para realizar una selección automática de variables en nuestro modelo de regresión logística. Este procedimiento parte de un modelo completo con todas las variables, y mediante un proceso llamado selección hacia adelante y hacia atrás (`direction = "both"`), elimina o incluye variables en función del criterio AIC (Criterio de Información de Akaike), buscando obtener el modelo más eficiente sin variables innecesarias. Esto nos ayuda a simplificar el modelo, manteniendo solo aquellas variables que realmente aportan a la predicción de la enfermedad cardíaca.

```
modelo_step <- step(glm(Enfermedad_Cardíaca~ ., family = binomial, data = enfermedad_train), direction = "both")
```

```

## Start: AIC=514.37
## Enfermedad_Cardíaca ~ Edad + Sexo + Dolor_Pecho + Presión_Reposo +
## Colesterol + Glucosa_Ayunas + Electrocardiograma_Reposo +
## Frecuencia_Máxima + Angina_Ejercicio + Depresión_ST + Pendiente_ST
##
##               Df Deviance   AIC
## - Presión_Reposo      1   482.42 512.42
## - Electrocardiograma_Reposo 2   484.97 512.97
## - Edad                1   483.36 513.36
## <none>                1   482.37 514.37
## - Frecuencia_Máxima    1   484.59 514.59
## - Depresión_ST         1   489.49 519.49
## - Colesterol           1   490.37 520.37
## - Angina_Ejercicio     1   491.63 521.63
## - Glucosa_Ayunas       1   493.77 523.77
## - Sexo                 1   507.96 537.96
## - Dolor_Pecho          3   531.46 557.46
## - Pendiente_ST        2   572.03 600.03
##
## Step: AIC=512.42
## Enfermedad_Cardíaca ~ Edad + Sexo + Dolor_Pecho + Colesterol +
## Glucosa_Ayunas + Electrocardiograma_Reposo + Frecuencia_Máxima +
## Angina_Ejercicio + Depresión_ST + Pendiente_ST
##
##               Df Deviance   AIC
## - Electrocardiograma_Reposo 2   485.04 511.04
## - Edad                1   483.36 511.36
## <none>                1   482.42 512.42
## - Frecuencia_Máxima    1   484.63 512.63
## + Presión_Reposo       1   482.37 514.37
## - Depresión_ST         1   489.49 517.49
## - Colesterol           1   490.95 518.95
## - Angina_Ejercicio     1   491.63 519.63
## - Glucosa_Ayunas       1   493.80 521.80
## - Sexo                 1   508.05 536.05
## - Dolor_Pecho          3   532.09 556.09
## - Pendiente_ST        2   572.15 598.15
##
## Step: AIC=511.04
## Enfermedad_Cardíaca ~ Edad + Sexo + Dolor_Pecho + Colesterol +
## Glucosa_Ayunas + Frecuencia_Máxima + Angina_Ejercicio +
## Depresión_ST + Pendiente_ST
##
##               Df Deviance   AIC
## - Frecuencia_Máxima    1   486.34 510.34
## - Edad                1   486.76 510.76
## <none>                1   485.04 511.04
## + Electrocardiograma_Reposo 2   482.42 512.42
## + Presión_Reposo       1   484.97 512.97
## - Colesterol           1   492.21 516.21
## - Depresión_ST         1   492.29 516.29
## - Angina_Ejercicio     1   493.52 517.52
## - Glucosa_Ayunas       1   495.95 519.95
## - Sexo                 1   510.78 534.78
## - Dolor_Pecho          3   535.22 555.22
## - Pendiente_ST        2   577.14 599.14
##
## Step: AIC=510.34
## Enfermedad_Cardíaca ~ Edad + Sexo + Dolor_Pecho + Colesterol +
## Glucosa_Ayunas + Angina_Ejercicio + Depresión_ST + Pendiente_ST
##
##               Df Deviance   AIC
## <none>                1   486.34 510.34
## + Frecuencia_Máxima    1   485.04 511.04
## - Edad                1   489.48 511.48
## + Presión_Reposo       1   486.29 512.29
## + Electrocardiograma_Reposo 2   484.63 512.63
## - Depresión_ST         1   492.89 514.89
## - Colesterol           1   495.62 517.62
## - Angina_Ejercicio     1   496.75 518.75

```



```
## - Glucosa_Ayunas      1  497.05 519.05
## - Sexo                1  512.68 534.68
## - Dolor_Pecho         3  539.74 557.74
## - Pendiente_ST       2  586.59 606.59
```

```
summary(modelo_step)
```

```
##
## Call:
## glm(formula = Enfermedad_Cardíaca ~ Edad + Sexo + Dolor_Pecho +
##      Colesterol + Glucosa_Ayunas + Angina_Ejercicio + Depresión_ST +
##      Pendiente_ST, family = binomial, data = enfermedad_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.139067    0.959594  -1.187  0.235216
## Edad           0.022857    0.012915   1.770  0.076753 .
## SexoM         1.509953    0.304802   4.954  7.28e-07 ***
## Dolor_PechoATA -1.770548    0.359945  -4.919  8.70e-07 ***
## Dolor_PechoNAP -1.715847    0.294574  -5.825  5.72e-09 ***
## Dolor_PechoTA  -1.802923    0.487220  -3.700  0.000215 ***
## Colesterol    -0.003445    0.001157  -2.978  0.002904 **
## Glucosa_Ayunas 0.975918    0.305241   3.197  0.001388 **
## Angina_Ejercicio 0.863893    0.267546   3.229  0.001242 **
## Depresión_ST   0.322509    0.128134   2.517  0.011837 *
## Pendiente_STFlat 0.897922    0.517184   1.736  0.082533 .
## Pendiente_STUp -1.613669    0.544351  -2.964  0.003033 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 1008.78  on 735  degrees of freedom
## Residual deviance: 486.34  on 724  degrees of freedom
## AIC: 510.34
##
## Number of Fisher Scoring iterations: 5
```

Obtuvimos un modelo más compacto y eficiente, que mantiene las variables clínicamente más relevantes, como el tipo de dolor en el pecho, los niveles de colesterol y glucosa, y la presencia de angina inducida por ejercicio.

- **SexoM:** Ser hombre incrementa significativamente el riesgo de enfermedad cardíaca. Esto se explica por el mayor riesgo cardiovascular que presentan los hombres en edad media, debido en parte a la protección hormonal que tienen las mujeres antes de la menopausia.
- **Dolor_Pecho:** El tipo de dolor en el pecho es uno de los indicadores clínicos más fuertes. El dolor típicamente anginoso (ASY) está más estrechamente relacionado con enfermedad coronaria, mientras que los dolores atípicos (ATA, NAP, TA) tienen menor especificidad diagnóstica.
- **Glucosa_Ayunas:** La presencia de glucosa elevada en ayunas es un claro marcador de alteraciones metabólicas como la diabetes o la resistencia a la insulina, condiciones altamente asociadas a eventos cardiovasculares.
- **Colesterol:** Aunque clínicamente se asocia un mayor nivel de colesterol con mayor riesgo cardiovascular, en este modelo su coeficiente resultó negativo. Esto podría explicarse por la presencia de pacientes tratados o con valores anómalos que distorsionan esta relación, o porque otras variables explican mejor el riesgo.
- **Angina_Ejercicio:** La presencia de angina inducida por el ejercicio es un signo de isquemia cardíaca, lo que la convierte en un fuerte predictor de enfermedad coronaria.
- **Depresión_ST:** Cada unidad adicional en la depresión del segmento ST refleja un mayor grado de isquemia durante el esfuerzo. Este hallazgo es coherente con lo observado en pruebas de esfuerzo y tiene alta relevancia clínica.
- **Pendiente_ST:** La morfología del segmento ST es un indicador clave en los estudios electrocardiográficos. Una pendiente “plana” o descendente suele asociarse a mayor riesgo, mientras que una pendiente “ascendente” (Up) generalmente se considera normal.

Interpretación de Coeficientes

En términos de Log-odds y odds.

- **SexoM (1.5099)(4.53) :** Ser hombre multiplica por 4.53 las odds de presentar enfermedad cardíaca.
- **Dolor_Pecho (-1.7705,-1.7158,-1.8029)(0.17,0.18,0.16):** Estos tipos de dolor reducen las odds de enfermedad cardíaca en más del 80% respecto al tipo de referencia (ASY).

- Colesterol(-0.0034)(0.9966): Cada unidad adicional de colesterol reduce las odds en 0.9966%.
- Glucosa_AyunasSí (0.9759)(2.65): Si tiene glucosa en ayunas alta, las odds aumentan 2.65 veces.
- Angina_EjercicioSí (0.8639)(2.37) : Tener angina con ejercicio duplica las odds de enfermedad cardíaca.
- Depresión_ST (0.3225)(1.38) : Cada unidad más de depresión ST aumenta las odds en 38%.
- Pendiente_ST (0.8979,-1.6137)(2.45,0.20) : Una pendiente plana multiplica por 2.45 las odds de enfermedad, mientras que una pendiente ascendente reduce las odds en 80%.

Matriz de Confusión

Esto permite interpretar mejor los factores de riesgo sin sacrificar capacidad predictiva. Que lo veremos a continuación

```
# Paso 1: Calcular las probabilidades para los datos de prueba
probabilidades <- predict(modelo_logistico,newdata=enfermedad_test,type="response")
head(probabilidades)
```

```
##           1           2           4           7           9           25
## 0.99585363 0.90040092 0.06354911 0.88505106 0.03776027 0.64845292
```

```
# Paso 2: Convertir probabilidades en clases 0 o 1 con umbral 0.5
predicciones_clase <- ifelse(probabilidades < 0.5,"Sin Enfermedad","Con Enfermedad")
head(predicciones_clase)
```

```
##           1           2           4           7
## "Con Enfermedad" "Con Enfermedad" "Sin Enfermedad" "Con Enfermedad"
##           9           25
## "Sin Enfermedad" "Con Enfermedad"
```

```
# Hay que asegurarse que ambas estén como factores
y_real<-enfermedad_test$Enfermedad_Cardíaca
y_pred<-factor(predicciones_clase,levels=c("Sin Enfermedad","Con Enfermedad"))

# Matriz de confusión y métricas
conf_matrix <- confusionMatrix(data=y_pred,reference=y_real,positive="Con Enfermedad")
print(conf_matrix$table)
```

```
##           Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad          71           8
## Con Enfermedad          17          86
```

```
# Métricas individuales
accuracy <- conf_matrix$overall["Accuracy"] # Proporción total de predicciones correctas.
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8626374
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.8349515
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.9148936
```

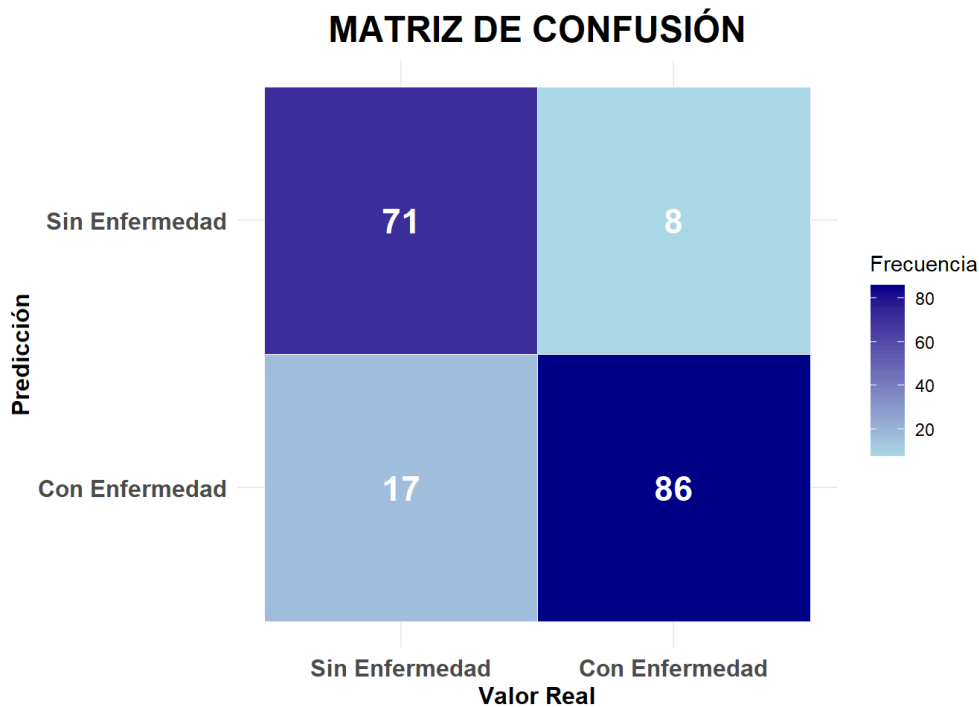
```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.8730964
```

```
cm_table <- as.data.frame(conf_matrix$table)

# Invertir el orden de los niveles del eje Y
cm_table$Prediction <- factor(cm_table$Prediction, levels = rev(levels(cm_table$Prediction)))

ggplot(cm_table, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), color = "white", size = 6, fontface = "bold") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name="Frecuencia") +
  labs(
    title = "MATRIZ DE CONFUSIÓN",
    x = "Valor Real",
    y = "Predicción"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    axis.text = element_text(size = 12, face="bold"),
    axis.title = element_text(size = 12, face="bold")
  )
```



Se obtuvieron las mismas métricas que en el primer modelo logístico, por lo que la interpretación es la misma.

ANÁLISIS DISCRIMINANTE LINEAL (LDA) O CUADRÁTICO (QDA)

Para empezar el modelado de nuestro LDA O QDA, primero tenemos que separar nuestras variables numéricas, haciendo caso omiso por el momento de las categóricas, así mismo esto lo haremos para train y test.

```
# VARIABLES NUMÉRICAS
enfermedad_train_num<-data.frame(select_if(enfermedad_train,is.numeric),
                                  Enfermedad_Cardíaca=enfermedad_train$Enfermedad_Cardíaca)
enfermedad_test_num<-data.frame(select_if(enfermedad_test,is.numeric),
                                 Enfermedad_Cardíaca=enfermedad_test$Enfermedad_Cardíaca)
```

Ahora sí se da paso a la verificación de los 4 supuestos.

Normalidad Univariada

Verificar la normalidad de los datos para cada variable en cada grupo y la normalidad multivariada.

```
vars <- names(enfermedad_train_num)[-6] # Excluye la columna de la clase
sapply(vars, function(var) {
  # Separación por clases para verificar normalidad
  tapply(enfermedad_train_num[[var]], enfermedad_train_num$Enfermedad_Cardíaca, function(x){
    p <- lillie.test(x)$p.value
    if (p > 0.05) {
      return(paste0("p = ", round(p, 4), " → Normal"))
    } else {
      return(paste0("p = ", round(p, 4), " → No normal"))
    }
  })
})
```

```
##           Edad           Presión_Reposo      Colesterol
## Sin Enfermedad "p = 0.0278 → No normal" "p = 0 → No normal" "p = 0 → No normal"
## Con Enfermedad "p = 0 → No normal"      "p = 0 → No normal" "p = 0 → No normal"
##           Frecuencia_Máxima      Depresión_ST
## Sin Enfermedad "p = 6e-04 → No normal" "p = 0 → No normal"
## Con Enfermedad "p = 0.016 → No normal" "p = 0 → No normal"
```

Aquí estamos aplicando el test de Lilliefors para cada variable, separando por dos grupos: - Sin enfermedad cardíaca - Con enfermedad cardíaca. Para ver si la distribución de cada variable es normal en cada grupo:

En este caso, todas las variables tienen p menor al nivel de significancia, por lo tanto en el contexto de LDA (Linear Discriminant Analysis), esta es una señal de que los supuestos de normalidad no se cumplen.

Ahora vamos con la normalidad multivariada (que es más importante para LDA).

Normalidad Multivariada

Se utilizó el test de Henze-Zirkler (HZ) debido a que es una prueba robusta ante valores atípicos y por el tamaño de los datos.

```
mvn(data=subset(enfermedad_train_num, Enfermedad_Cardíaca=="Sin Enfermedad"), -6,
     mvn_test="hz")
```

```
## $multivariate_normality
##           Test Statistic p.value           MVN
## 1 Henze-Zirkler      3.154 <0.001 X Not normal
##
## $univariate_normality
##           Test           Variable Statistic p.value      Normality
## 1 Anderson-Darling      Edad      0.640  0.095    ✓ Normal
## 2 Anderson-Darling  Presión_Reposo  4.060 <0.001 X Not normal
## 3 Anderson-Darling      Colesterol  7.028 <0.001 X Not normal
## 4 Anderson-Darling  Frecuencia_Máxima 1.623 <0.001 X Not normal
## 5 Anderson-Darling      Depresión_ST 46.500 <0.001 X Not normal
##
## $descriptives
##           Variable   n    Mean Std.Dev Median  Min   Max   25th   75th   Skew
## 1           Edad 322  50.550  9.372   51.0 28.0  76.0  43.00  57.00  0.057
## 2   Presión_Reposo 322 130.969 16.670  130.0 94.0 190.0 120.00 140.00  0.709
## 3      Colesterol 322 227.478 66.606  226.5  0.0 412.0 199.00 264.75 -1.061
## 4 Frecuencia_Máxima 322 148.385 23.748  150.0 80.0 202.0 134.25 166.00 -0.415
## 5      Depresión_ST 322  0.404  0.703   0.0 -0.1  4.2  0.00  0.60  2.023
##
## Kurtosis
## 1      2.618
## 2      3.810
## 3      6.431
## 4      2.653
## 5      7.412
##
## $data
##      Edad Presión_Reposo Colesterol Frecuencia_Máxima Depresión_ST
## 3      37          130          194             150          0.0
## 5      41          125          184             180          0.0
## 10     43          120          266             118          0.0
## 16     54          108          267             167          0.0
## 18     49          110          208             160          0.0
## 22     35          138          183             182          1.4
## 23     36          130          209             178          0.0
## 26     35          122          192             174          0.0
## 27     41          112          250             179          0.0
## 31     53          155          175             160          0.3
## 34     58          150          283             162          1.0
## 38     65          120          177             140          0.4
## 41     49          124          201             164          0.0
## 43     61          130          294             120          1.0
## 44     52          152          298             178          1.2
## 45     33          120          298             185          0.0
## 48     36          120          166             180          0.0
## 50     44          130          215             135          0.0
## 51     37          118          240             165          1.0
## 56     40          130          275             150          0.0
## 60     54          150          230             130          0.0
## 62     55          140          229             110          0.5
## 64     48          120          195             125          0.0
## 67     51          114          258              96          1.0
## 68     56          120          193             162          1.9
## 70     59          154           0             131          1.5
## 72     48          120          177             148          0.0
## 74     67          152          277             172          0.0
## 75     28          130          132             185          0.0
## 76     55          132          342             166          1.2
## 79     63          130           0             160          3.0
## 80     39          120          204             145          0.0
## 82     57          132          207             168          0.0
## 88     48          110          211             138          0.0
## 89     60          180           0             140          1.5
## 90     51          110          190             120          0.0
## 91     41          105          198             168          0.0
## 93     39          190          241             106          0.0
## 94     49          140          185             130          0.0
## 97     51          132          218             139          0.1
## 100    45          132          297             144          0.0
## 104    37          120          215             170          0.0
```

## 106	44	120	218	115	0.0
## 107	59	140	287	150	0.0
## 109	39	120	200	160	1.0
## 113	34	130	161	190	0.0
## 118	52	134	201	158	0.8
## 122	64	128	263	105	0.2
## 123	60	102	318	160	0.0
## 124	39	130	215	120	0.0
## 126	58	140	385	135	0.3
## 127	50	120	219	158	1.6
## 130	41	120	291	160	0.0
## 132	54	150	195	122	0.0
## 134	45	138	236	152	0.2
## 135	54	130	253	155	0.0
## 136	41	150	171	128	1.5
## 149	62	160	193	116	0.0
## 151	63	145	233	150	2.3
## 153	53	138	234	160	0.0
## 160	42	120	196	150	0.0
## 165	52	160	196	165	0.0
## 171	59	140	221	164	0.0
## 184	61	140	298	120	0.0
## 185	48	150	227	130	1.0
## 188	57	130	308	98	1.0
## 196	41	130	245	150	0.0
## 198	54	140	309	140	0.0
## 199	48	133	308	156	2.0
## 202	53	142	226	111	0.0
## 206	56	130	184	100	0.0
## 207	63	135	252	172	0.0
## 209	34	118	182	174	0.0
## 219	57	180	347	126	0.8
## 220	47	130	235	145	2.0
## 223	54	120	230	140	0.0
## 224	54	120	217	137	0.0
## 226	72	160	0	114	1.6
## 229	58	130	197	131	0.6
## 235	43	122	213	165	0.2
## 236	62	124	209	163	0.0
## 237	66	110	213	99	1.3
## 238	41	126	306	163	0.0
## 240	47	160	263	174	0.0
## 241	50	110	254	159	0.0
## 242	60	120	178	96	0.0
## 245	54	108	309	156	0.0
## 246	55	120	256	137	0.0
## 251	52	130	180	140	1.5
## 252	61	150	243	137	1.0
## 256	38	138	175	173	0.0
## 260	42	140	226	178	0.0
## 269	42	134	240	160	0.0
## 271	68	118	277	151	1.0
## 274	58	140	179	160	0.0
## 276	41	112	250	142	0.0
## 278	40	106	240	80	0.0
## 281	58	105	240	154	0.6
## 284	41	130	204	172	1.4
## 290	59	130	188	124	1.0
## 293	59	140	169	140	0.0
## 295	47	140	257	135	1.0
## 298	68	134	254	151	0.0
## 301	63	140	195	179	0.0
## 303	45	128	308	170	0.0
## 307	46	150	163	116	0.0
## 309	51	125	188	145	0.0
## 310	52	128	205	184	0.0
## 313	52	118	186	190	0.0
## 314	49	130	207	135	0.0
## 318	29	130	204	202	0.0
## 323	43	120	249	176	0.0
## 326	58	140	211	165	0.0

## 328	53	130	246	173	0.0
## 329	57	140	260	140	0.0
## 337	52	172	199	162	0.5
## 339	54	120	273	150	1.5
## 341	58	100	248	122	1.0
## 343	51	128	0	107	0.0
## 345	42	130	180	150	0.0
## 349	31	100	219	150	0.0
## 351	53	128	216	115	0.0
## 354	52	125	272	139	0.0
## 358	51	150	200	120	0.5
## 361	50	140	216	170	0.0
## 363	41	125	269	144	0.0
## 364	34	150	214	168	0.0
## 366	48	120	284	120	0.0
## 367	71	110	265	130	0.0
## 368	52	108	233	147	0.1
## 372	46	101	197	156	0.0
## 373	50	120	244	162	1.1
## 376	54	110	208	142	0.0
## 380	69	160	234	131	0.1
## 381	54	110	214	158	1.6
## 384	54	120	221	138	1.0
## 386	55	140	196	150	0.0
## 388	35	120	160	185	0.0
## 395	70	156	245	143	0.0
## 396	46	138	243	152	0.0
## 399	51	140	308	142	1.5
## 400	46	130	238	90	0.0
## 407	35	120	308	180	0.0
## 410	49	134	271	162	0.0
## 419	53	130	197	152	1.2
## 422	41	120	157	182	0.0
## 424	37	130	250	187	3.5
## 425	50	120	328	110	1.0
## 427	65	155	0	154	1.0
## 428	71	112	149	125	1.6
## 429	44	130	209	127	0.0
## 433	34	118	210	192	0.7
## 439	39	138	220	152	0.0
## 443	43	100	223	142	0.0
## 447	51	110	175	123	0.6
## 449	46	120	230	150	0.0
## 454	57	130	207	96	1.0
## 455	51	132	227	138	0.2
## 457	48	108	163	175	2.0
## 461	36	112	340	184	1.0
## 462	53	120	195	140	0.0
## 466	49	120	297	132	1.0
## 467	35	123	161	153	-0.1
## 472	46	180	280	120	0.0
## 476	41	110	235	153	0.0
## 478	54	150	365	134	1.0
## 480	43	130	315	162	1.9
## 481	49	100	253	174	0.0
## 489	53	140	216	142	2.0
## 491	49	130	269	163	0.0
## 494	42	160	147	146	0.0
## 496	56	130	221	163	0.0
## 500	38	145	292	130	0.0
## 501	62	128	208	140	0.0
## 504	54	132	288	159	0.0
## 505	48	140	238	118	0.0
## 506	59	180	213	100	0.0
## 508	65	155	269	148	0.8
## 510	39	110	182	180	0.0
## 515	46	110	238	140	1.0
## 517	60	152	0	118	0.0
## 520	54	150	232	165	1.6
## 526	58	130	251	110	0.0
## 532	52	140	100	138	0.0

## 537	57	120	354	163	0.6
## 539	48	130	245	180	0.2
## 541	60	136	195	126	0.3
## 542	56	120	240	169	0.0
## 543	39	120	241	146	2.0
## 544	56	120	0	97	0.0
## 549	40	130	281	167	0.0
## 550	43	115	303	181	1.2
## 553	54	140	239	160	1.2
## 557	47	110	249	150	0.0
## 562	48	122	222	186	0.0
## 563	50	110	202	145	0.0
## 566	50	150	215	140	0.0
## 568	47	112	204	143	0.1
## 577	59	150	212	157	1.6
## 581	45	110	0	138	-0.1
## 582	45	120	225	140	0.0
## 590	55	122	320	155	0.0
## 592	66	160	228	138	2.3
## 594	44	120	226	169	0.0
## 595	41	130	214	168	2.0
## 597	55	130	262	155	0.0
## 598	59	140	274	154	2.0
## 600	59	130	318	120	1.0
## 601	49	130	266	171	0.6
## 606	29	140	263	170	0.0
## 607	67	106	223	142	0.3
## 609	39	140	321	182	0.0
## 610	42	120	240	194	0.8
## 611	45	180	295	180	0.0
## 615	58	135	222	100	0.0
## 618	57	150	168	174	1.6
## 619	62	135	139	137	0.2
## 621	55	120	220	134	0.0
## 626	45	115	260	185	0.0
## 629	51	160	194	170	0.0
## 631	39	94	199	179	0.0
## 633	51	94	227	154	0.0
## 635	32	110	225	184	0.0
## 638	45	140	224	122	0.0
## 640	37	130	173	184	0.0
## 647	63	136	165	133	0.2
## 649	53	140	320	162	0.0
## 651	54	160	305	175	0.0
## 653	40	140	235	188	0.0
## 658	53	130	182	148	0.0
## 660	59	178	270	145	4.2
## 662	32	105	198	165	0.0
## 667	56	130	219	164	0.0
## 670	52	120	210	148	0.0
## 677	44	108	141	175	0.6
## 679	76	140	197	116	1.1
## 683	52	138	223	169	0.0
## 686	62	140	394	157	1.2
## 692	51	125	245	166	2.4
## 693	38	140	297	150	0.0
## 695	56	120	85	140	0.0
## 701	43	110	211	161	0.0
## 703	46	110	240	140	0.0
## 707	54	135	304	170	0.0
## 712	51	130	179	100	0.0
## 714	56	130	276	128	1.0
## 717	51	120	295	157	0.6
## 718	44	140	235	180	0.0
## 719	42	115	211	137	0.0
## 723	53	130	264	143	0.4
## 725	43	120	201	165	0.0
## 731	45	140	224	144	0.0
## 733	48	130	275	139	0.2
## 735	64	170	227	155	0.6
## 736	55	130	394	150	0.0


```

## 737 45      135      192      110      0.0
## 739 52      140      259      170      0.0
## 743 51      130      220      160      2.0
## 744 57      128      303      159      0.0
## 748 54      120      238      154      0.0
## 749 43      150      254      175      0.0
## 752 48      120      254      110      0.0
## 753 39      110      273      132      0.0
## 758 56      124      224      161      2.0
## 759 42      120      295      162      0.0
## 763 74      138        0      116      0.2
## 764 50      129      196      163      0.0
## 766 61      125      292      115      0.0
## 769 58      120      340      172      0.0
## 772 51      130      224      150      0.0
## 777 64      130      223      128      0.5
## 781 62      140      271      152      1.0
## 782 43      150      247      171      1.5
## 783 66      150      226      114      2.6
## 787 42      148      244      178      0.8
## 790 74      120      269      121      0.2
## 792 53      140      243      155      0.0
## 795 44      120      220      170      0.0
## 796 42      150      268      136      0.0
## 797 51      130      256      149      0.5
## 799 57      110      201      126      1.5
## 802 44      150      412      170      0.0
## 803 64      110      211      144      1.8
## 814 64      180      325      154      0.0
## 816 41      104        0      111      0.0
## 818 48      124      255      175      0.0
## 819 35      150      264      168      0.0
## 826 55      150      160      150      0.0
## 828 65      160      360      151      0.8
## 832 45      130      234      175      0.6
## 835 43      150      186      154      0.0
## 837 65      140      252      135      0.3
## 838 50      140      129      135      0.0
## 841 64      140      313      133      0.2
## 845 37      120      223      168      0.0
## 847 52      120      284      118      0.0
## 849 56      126      166      140      0.0
## 850 44      130      219      188      0.0
## 852 59      138      271      182      0.0
## 860 52      120      325      172      0.2
## 861 55      120      270      140      0.0
## 865 50      170      209      116      0.0
## 867 66      146      278      152      0.0
## 872 56      130      167      114      0.0
## 873 53      124      260      112      3.0
## 874 62      120      220      86      0.0
## 875 54      125      273      152      0.5
## 878 43      142      207      138      0.0
## 882 54      133      203      137      0.2
## 883 42      140      358      170      0.0
## 885 41      112      268      172      0.0
## 888 68      139      181      135      0.2
## 890 37      130      283      98      0.0
## 896 60      120        0      133      2.0
## 899 62      120      220      86      0.0
## 901 50      120      168      160      0.0
## 904 59      135      234      161      0.5
## 905 54      160      201      163      0.0
## 908 30      170      237      170      0.0
## 909 37      120      260      130      0.0
## 910 69      140      239      151      1.8
## 913 54      120      246      110      0.0
##
## $subset
## NULL
##

```

```
## $outlierMethod
## [1] "none"
##
## attr("class")
## [1] "mvn"
```

```
mvn(data=subset(enfermedad_train_num, Enfermedad_Cardíaca=="Con Enfermedad"), -6,
     mvn_test="hz", multivariate_outlier_method="quan")
```

```
## $multivariate_normality
##           Test Statistic p.value      MVN
## 1 Henze-Zirkler      2.285 <0.001 X Not normal
##
## $univariate_normality
##           Test      Variable Statistic p.value      Normality
## 1 Anderson-Darling      Edad      2.330 <0.001 X Not normal
## 2 Anderson-Darling  Presión_Reposo      2.816 <0.001 X Not normal
## 3 Anderson-Darling      Colesterol      26.975 <0.001 X Not normal
## 4 Anderson-Darling  Frecuencia_Máxima      0.535      0.17      ✓ Normal
## 5 Anderson-Darling      Depresión_ST      7.712 <0.001 X Not normal
##
## $descriptives
##           Variable   n   Mean Std.Dev Median Min   Max 25th 75th  Skew
## 1           Edad 414  55.829   8.941   57.0  31  77.0   51   62 -0.385
## 2   Presión_Reposo 414 133.486  19.918  132.0   0 200.0  120  144 -0.200
## 3      Colesterol 414 175.203 126.225  217.0   0 529.0   0  268 -0.299
## 4 Frecuencia_Máxima 414 127.882  23.493  126.0  60 195.0  112  145 -0.041
## 5      Depresión_ST 414   1.254   1.162   1.2  -2   6.2   0   2  0.669
##
## Kurtosis
## 1      2.967
## 2      7.787
## 3      1.988
## 4      2.773
## 5      3.765
##
## $multivariate_outliers
##           Observation Mahalanobis.Distance
## 1              289             83.101
## 2              578             46.783
## 3              344             39.165
## 4              710             39.000
## 5              129             38.372
## 6              672             37.652
## 7              691             36.012
## 8              839             35.997
## 9              659             35.383
## 10             536             35.309
## 11             561             35.276
## 12             181             35.226
## 13             286             35.151
## 14             108             34.775
## 15             815             33.833
## 16             572             33.565
## 17             423             33.421
## 18             304             33.339
## 19             133             33.329
## 20             622             33.294
## 21             811             33.228
## 22             327             32.742
## 23             333             32.451
## 24             709             32.407
## 25             511             32.357
## 26             117             32.204
## 27             576             31.927
## 28             194             31.917
## 29             634             31.745
## 30             273             31.730
## 31             756             31.609
## 32             856             31.509
## 33             365             31.456
## 34             247             31.404
## 35             353             31.147
## 36             593             31.081
## 37             503             30.965
## 38             436             30.738
## 39             336             30.625
## 40             387             30.412
## 41             216             30.367
## 42             708             30.271
```

## 43	232	30.125
## 44	8	30.061
## 45	168	29.953
## 46	434	29.858
## 47	862	29.736
## 48	789	29.440
## 49	617	29.173
## 50	512	29.155
## 51	266	29.062
## 52	479	29.049
## 53	637	28.877
## 54	464	28.796
## 55	57	28.725
## 56	83	28.693
## 57	812	28.657
## 58	317	28.609
## 59	412	28.580
## 60	521	28.552
## 61	587	28.373
## 62	297	28.296
## 63	24	28.235
## 64	21	28.145
## 65	854	27.875
## 66	727	27.828
## 67	778	27.751
## 68	548	27.671
## 69	218	27.614
## 70	63	27.499
## 71	157	27.467
## 72	751	27.266
## 73	696	27.250
## 74	211	27.181
## 75	809	26.844
## 76	19	26.801
## 77	605	26.745
## 78	139	26.708
## 79	715	26.661
## 80	155	26.654
## 81	214	26.624
## 82	846	26.573
## 83	458	26.546
## 84	915	26.503
## 85	669	26.476
## 86	632	26.448
## 87	786	26.402
## 88	362	26.341
## 89	291	26.341
## 90	726	26.235
## 91	664	26.218
## 92	791	26.114
## 93	322	26.002
## 94	858	25.992
## 95	331	25.974
## 96	431	25.964
## 97	420	25.908
## 98	716	25.872
## 99	829	25.854
## 100	513	25.833
## 101	694	25.795
## 102	222	25.750
## 103	900	25.717
## 104	493	25.713
## 105	201	25.689
## 106	180	25.603
## 107	389	25.600
## 108	285	25.580
## 109	655	25.543
## 110	652	25.529
## 111	487	25.478
## 112	754	25.315
## 113	177	25.264

```

## 114      720      25.256
## 115      379      25.175
## 116       61      25.151
## 117      738      25.144
## 118      398      25.035
## 119      258      25.022
## 120      406      24.952
## 121      773      24.935
## 122      869      24.933
## 123       42      24.855
## 124       12      24.847
## 125      750      24.569
## 126      174      24.534
## 127      394      24.508
## 128       87      24.416
## 129      347      24.415
## 130      111      24.245
## 131      173      23.559
## 132      259      21.815
## 133      884      20.502
## 134      518      19.578
## 135      203      18.451
## 136      868      17.037
## 137      440      16.445
## 138      547      14.938
## 139      567      14.644
## 140      250      14.547
## 141      827      14.531
## 142      574      12.995
##
## $data
##      Edad Presión_Reposo Colesterol Frecuencia_Máxima Depresión_ST
## 6      60          125         258             141         2.8
## 8      51          110           0             92         0.0
## 11     49          140          234            140         1.0
## 12     59          135           0            115         1.0
## 13     65          150          225            114         1.0
## 14     61          140          284            123         1.3
## 15     61          120          337             98         0.0
## 17     54          110          206            108         0.0
## 19     48          115           0            128         0.0
## 20     59          170          326            140         3.4
## 21     55          120           0             92         0.3
## 24     48          102           0            110         1.0
## 28     48          106          263            110         0.0
## 29     74          140          237             94         0.0
## 35     47          160          291            158         3.0
## 36     70          170          192            129         3.0
## 39     66          178          228            165         1.0
## 40     62          130          263             97         1.2
## 42     58          115           0            138         0.5
## 46     46          140          272            175         2.0
## 47     39          110          280            150         0.0
## 52     69          142          210            112         1.5
## 53     52          160          331             94         2.5
## 55     62          140          268            160         3.6
## 57     65          115           0             93         0.0
## 58     51          135          160            150         2.0
## 59     57          130          236            174         0.0
## 61     57          128           0            148         1.0
## 63     74          145           0            123         1.3
## 66     33          100          246            150         1.0
## 69     66          112          212            132         0.1
## 73     44          130          290            100         2.0
## 77     57          140          241            123         0.2
## 83     59          110           0             94         0.0
## 85     52          170          223            126         1.5
## 87     62          133           0            119         1.2
## 92     47          135          248            170         0.0
## 95     46          120          277            125         1.0
## 99     47          110          275            118         1.0

```

## 101	44	110	197
## 105	50	130	233
## 108	47	160	0
## 110	46	134	310
## 111	60	130	0
## 112	55	180	327
## 114	50	140	288
## 115	61	120	260
## 116	65	138	282
## 117	57	95	0
## 119	55	160	292
## 120	51	140	298
## 125	42	136	315
## 129	51	140	0
## 131	57	154	232
## 133	42	105	0
## 137	55	140	295
## 138	58	132	224
## 139	63	140	0
## 140	39	118	219
## 142	58	136	203
## 143	57	152	274
## 144	64	134	273
## 145	67	140	219
## 152	58	128	216
## 154	62	135	297
## 155	52	130	0
## 156	56	132	184
## 157	61	134	0
## 158	36	120	267
## 159	45	110	264
## 161	47	120	205
## 162	63	110	252
## 163	54	140	216
## 164	74	145	216
## 166	50	144	349
## 167	53	140	203
## 168	70	115	0
## 170	57	122	264
## 172	56	134	409
## 173	62	160	164
## 174	63	133	0
## 175	52	125	212
## 177	59	140	0
## 179	49	130	341
## 180	62	110	0
## 181	58	170	0
## 186	61	120	282
## 190	47	140	193
## 191	58	160	211
## 192	64	150	193
## 194	40	125	0
## 197	74	150	258
## 201	56	125	0
## 203	56	200	288
## 204	60	130	206
## 205	60	150	258
## 210	54	110	239
## 211	69	135	0
## 212	62	138	294
## 214	67	120	0
## 215	31	120	270
## 216	61	120	0
## 218	46	115	0
## 221	57	128	229
## 222	59	120	0
## 225	57	124	261
## 228	54	130	202
## 230	67	125	254
## 231	71	144	221
## 232	61	160	0

177	0.0
121	2.0
124	0.0
126	0.0
130	1.1
117	3.4
140	0.0
140	3.6
174	1.4
182	0.7
143	2.0
122	4.2
125	1.8
60	0.0
164	0.0
128	-1.5
136	0.0
173	3.2
149	2.0
140	1.2
123	1.2
88	1.2
102	4.0
122	2.0
131	2.2
130	1.0
120	0.0
105	2.1
86	1.5
160	3.0
132	1.2
98	2.0
140	2.0
105	1.5
116	1.8
120	1.0
155	3.1
92	0.0
100	0.0
150	1.9
145	6.2
120	1.0
168	1.0
117	1.0
120	1.0
120	0.5
105	0.0
135	4.0
145	1.0
92	0.0
135	0.5
165	0.0
130	4.0
103	1.0
133	4.0
132	2.4
157	2.6
126	2.8
130	0.0
106	1.9
150	1.5
153	1.5
80	0.0
113	1.5
150	0.4
115	0.0
141	0.3
112	2.0
163	0.2
108	1.8
145	1.0

## 233	61	146	241	148	3.0
## 234	51	160	303	150	1.0
## 239	59	126	218	134	2.2
## 243	67	146	369	110	1.9
## 247	55	140	0	83	0.0
## 248	38	120	282	170	0.0
## 250	77	125	304	162	0.0
## 258	54	130	0	117	1.4
## 259	40	120	466	152	1.0
## 261	54	125	216	140	0.0
## 265	64	145	212	132	2.0
## 266	61	150	0	105	0.0
## 267	54	120	188	113	1.4
## 268	44	150	288	150	3.0
## 272	44	112	290	153	0.0
## 273	47	155	0	118	1.0
## 279	64	125	309	131	1.8
## 280	49	130	206	170	0.0
## 282	61	142	200	100	1.5
## 285	61	110	0	113	1.4
## 286	38	105	0	166	2.8
## 287	57	130	311	148	2.0
## 289	55	0	0	155	1.5
## 291	57	105	0	148	0.3
## 292	47	108	243	152	0.0
## 297	50	145	0	139	0.7
## 300	41	120	336	118	3.0
## 304	38	135	0	150	0.0
## 305	60	130	253	144	1.4
## 306	67	160	286	108	1.5
## 308	48	160	329	92	1.5
## 311	52	140	266	134	2.0
## 312	43	132	341	136	3.0
## 315	58	150	270	111	0.8
## 317	53	120	0	95	0.0
## 319	58	130	263	140	2.0
## 320	60	145	282	142	2.8
## 322	52	135	0	128	2.0
## 327	63	150	0	154	3.7
## 331	56	120	0	148	0.0
## 332	59	160	273	125	0.0
## 333	65	145	0	67	0.7
## 335	61	140	207	138	1.9
## 336	53	145	518	130	0.0
## 338	54	127	333	154	0.0
## 340	55	142	228	149	2.5
## 344	54	180	0	150	1.5
## 346	59	124	160	117	1.0
## 347	58	116	0	124	1.0
## 350	53	144	300	128	1.5
## 352	56	137	282	126	1.2
## 353	38	110	0	156	0.0
## 356	57	180	285	120	0.8
## 360	63	130	254	147	1.4
## 362	60	115	0	143	2.4
## 365	60	160	0	149	0.4
## 369	61	148	203	161	0.0
## 371	55	122	223	100	0.0
## 375	43	132	247	143	0.1
## 378	62	138	204	122	1.2
## 379	51	120	0	127	1.5
## 383	52	130	298	110	1.0
## 385	55	158	217	110	2.5
## 387	43	140	0	140	0.5
## 389	64	144	0	122	1.0
## 393	48	160	193	102	3.0
## 394	62	120	0	123	1.7
## 397	46	118	186	124	0.0
## 398	53	130	0	135	1.0
## 402	71	130	221	115	0.0
## 404	48	130	256	150	0.0

## 406	53	125	0
## 409	60	100	248
## 411	61	145	307
## 412	65	160	0
## 413	62	112	258
## 414	40	150	392
## 415	63	140	187
## 416	56	137	208
## 417	41	120	237
## 418	55	136	228
## 420	61	130	0
## 421	54	140	166
## 423	60	135	0
## 430	57	150	276
## 431	50	115	0
## 434	61	130	0
## 436	40	95	0
## 437	45	142	309
## 438	66	112	261
## 440	34	140	156
## 441	50	150	243
## 442	67	152	212
## 445	53	120	246
## 446	65	170	263
## 448	56	128	223
## 453	49	118	149
## 458	61	150	0
## 459	56	155	342
## 463	48	160	355
## 464	64	95	0
## 468	60	130	186
## 469	53	124	243
## 470	53	180	285
## 471	57	150	255
## 473	65	136	248
## 477	60	130	186
## 479	51	95	0
## 482	56	125	249
## 484	60	140	293
## 485	68	150	195
## 486	58	120	284
## 487	69	130	0
## 488	65	144	312
## 490	46	110	202
## 492	60	158	305
## 493	44	135	491
## 498	67	100	299
## 499	69	140	208
## 502	49	128	212
## 503	63	100	0
## 509	60	141	316
## 511	35	120	0
## 512	70	140	0
## 513	64	110	0
## 516	59	140	264
## 518	54	200	198
## 519	52	130	225
## 521	43	122	0
## 522	68	180	274
## 523	63	124	197
## 524	64	140	335
## 525	55	140	201
## 528	63	170	177
## 530	65	135	254
## 535	51	140	299
## 536	32	95	0
## 540	46	120	249
## 546	60	140	185
## 547	55	140	217
## 548	62	120	0
## 554	46	110	236

120	1.5
125	1.0
146	1.0
122	1.2
150	1.3
130	2.0
144	4.0
122	1.8
138	1.0
124	1.6
115	0.0
118	0.0
63	0.5
112	0.6
120	0.5
77	2.5
144	0.0
147	0.0
140	1.5
180	0.0
128	2.6
150	0.8
116	0.0
112	2.0
119	2.0
126	0.8
117	2.0
150	3.0
99	2.0
145	1.1
140	0.5
122	2.0
120	1.5
92	3.0
140	4.0
140	0.5
126	2.2
144	1.2
170	1.2
132	0.0
160	1.8
129	1.0
113	1.7
150	0.0
161	0.0
135	0.0
125	0.9
140	2.0
96	0.0
109	-0.9
122	1.7
130	1.2
157	2.0
114	1.3
119	0.0
142	2.0
120	2.0
120	0.5
150	1.6
136	0.0
158	0.0
130	3.0
84	2.5
127	2.8
173	1.6
127	0.7
144	0.8
155	3.0
111	5.6
134	-0.8
125	2.0

## 555	59	130	126	125	0.0
## 556	57	165	289	124	1.0
## 558	50	140	233	163	0.6
## 559	51	130	305	142	1.2
## 560	61	141	292	115	1.7
## 561	56	125	0	98	-2.0
## 565	53	123	282	95	2.0
## 567	58	180	393	110	1.0
## 569	46	120	231	115	0.0
## 572	56	115	0	82	-1.0
## 574	58	114	318	140	4.4
## 576	73	160	0	121	0.0
## 578	64	200	0	140	1.0
## 580	54	138	274	105	1.5
## 584	48	124	274	166	0.5
## 585	50	140	231	140	5.0
## 586	55	145	248	96	2.0
## 587	57	140	0	100	0.0
## 588	57	144	270	160	2.0
## 591	48	132	220	162	0.0
## 593	38	115	0	128	0.0
## 602	58	110	198	110	0.0
## 603	62	120	254	93	0.0
## 605	66	150	0	108	2.0
## 608	67	120	237	71	1.0
## 614	60	140	281	118	1.5
## 617	49	130	0	145	3.0
## 622	52	95	0	82	0.8
## 623	59	134	204	162	0.8
## 627	63	108	269	169	1.8
## 628	63	160	267	88	2.0
## 632	55	115	0	155	0.1
## 634	36	110	0	125	1.0
## 636	46	140	311	120	1.8
## 637	63	150	0	86	2.0
## 639	55	172	260	73	2.0
## 642	55	120	226	127	1.7
## 644	60	117	230	160	1.4
## 645	65	150	235	120	1.5
## 646	46	150	231	147	3.6
## 648	63	130	308	138	2.0
## 650	40	152	223	181	0.0
## 652	56	140	0	121	1.8
## 654	61	130	330	169	0.0
## 655	57	140	0	120	2.0
## 656	55	132	353	132	1.2
## 657	38	110	196	166	0.0
## 659	42	145	0	99	0.0
## 661	59	170	288	159	0.2
## 663	37	140	207	130	1.5
## 664	63	130	0	111	0.0
## 668	77	124	171	110	2.0
## 669	61	110	0	108	2.0
## 672	59	178	0	120	0.0
## 673	58	112	230	165	2.5
## 674	57	156	173	119	3.0
## 676	59	110	239	142	1.2
## 681	48	138	214	108	1.5
## 682	59	174	249	143	0.0
## 684	41	110	289	170	0.0
## 685	64	143	306	115	1.8
## 688	55	140	268	128	1.5
## 690	58	125	300	171	0.0
## 691	38	150	0	120	0.7
## 694	54	192	283	195	0.0
## 696	53	126	0	106	0.0
## 697	58	130	213	140	0.0
## 698	62	160	254	108	3.0
## 704	63	96	305	121	1.0
## 706	59	125	222	135	2.5
## 708	56	120	0	100	-1.0

## 709	53	160	0
## 710	32	118	529
## 713	52	128	255
## 715	54	120	0
## 716	55	120	0
## 720	58	120	0
## 726	52	122	0
## 727	51	120	0
## 728	54	130	294
## 729	58	136	319
## 730	54	124	266
## 732	52	120	182
## 734	64	142	276
## 738	58	126	0
## 741	54	120	237
## 742	55	128	205
## 745	63	139	217
## 750	56	130	0
## 751	50	120	0
## 754	51	128	0
## 755	56	130	283
## 756	56	155	0
## 761	69	140	254
## 765	60	132	218
## 768	75	170	203
## 773	65	134	0
## 774	62	120	267
## 775	56	130	256
## 778	67	145	0
## 780	38	110	289
## 785	57	130	131
## 786	69	140	0
## 788	45	130	219
## 789	43	100	0
## 791	62	115	0
## 793	70	145	174
## 794	65	150	236
## 798	55	136	245
## 800	72	160	123
## 801	41	110	172
## 808	56	130	203
## 809	68	135	0
## 810	69	145	289
## 811	34	115	0
## 812	43	115	0
## 815	62	115	0
## 823	58	100	234
## 824	66	160	246
## 825	44	120	169
## 827	38	120	231
## 829	58	130	0
## 830	63	160	230
## 834	43	150	247
## 836	52	112	230
## 839	62	160	0
## 840	58	170	225
## 842	52	140	404
## 843	55	116	186
## 844	49	131	142
## 846	68	145	0
## 848	70	160	269
## 853	40	110	167
## 854	47	110	0
## 855	54	130	242
## 856	66	155	0
## 857	68	144	193
## 858	53	120	0
## 859	60	142	216
## 862	63	136	0
## 863	35	110	257
## 868	63	150	407

122	0.0
130	0.0
161	0.0
155	0.0
125	2.5
106	1.5
110	2.0
104	0.0
100	0.0
152	0.0
109	2.2
150	0.0
140	1.0
110	2.0
150	1.5
130	2.0
128	1.2
122	1.0
156	0.0
125	1.2
103	1.6
99	0.0
146	2.0
140	1.5
108	0.0
112	1.1
99	1.8
142	0.6
125	0.0
105	1.5
115	1.2
118	2.5
130	1.0
122	1.5
128	2.5
125	2.6
105	0.0
131	1.2
130	1.5
158	0.0
98	1.5
120	0.0
110	1.8
154	0.2
145	2.0
72	-0.5
156	0.1
120	0.0
144	2.8
182	3.8
100	1.0
105	1.0
130	2.0
160	0.0
72	0.0
146	2.8
124	2.0
102	0.0
127	1.5
136	1.8
112	2.9
114	2.0
149	2.1
91	1.0
90	0.0
141	3.4
120	0.0
110	2.5
84	0.0
140	0.0
154	4.0

```
## 869 57 110 0 131 1.4
## 870 62 120 281 103 1.4
## 871 35 126 282 156 0.0
## 877 62 152 153 97 1.6
## 880 49 120 188 139 2.0
## 881 48 160 268 103 1.0
## 884 76 104 113 120 3.5
## 886 55 160 289 145 0.8
## 887 58 136 164 99 2.0
## 892 58 137 232 124 1.4
## 894 62 139 170 120 3.0
## 897 69 140 110 109 1.5
## 898 38 110 190 150 1.0
## 900 64 120 0 106 2.0
## 902 67 120 229 129 2.6
## 903 75 136 225 112 3.0
## 906 72 120 214 102 1.0
## 907 47 150 226 98 1.5
## 915 61 125 0 105 0.0
## 916 48 122 275 150 2.0
##
## $subset
## NULL
##
## $outlierMethod
## [1] "quan"
##
## attr(,"class")
## [1] "mvn"
```

Ahora vemos la normalidad multivariada con el test de Henze-Zirkler (HZ), y también viendo la normalidad univariada con el test de Anderson-Darling, ambos realizados por grupo (Sin Enfermedad y Con Enfermedad).

En el grupo sin enfermedad y con enfermedad vemos que se ocupa el test de Henze-Zirkler, donde nos arroja un p-value de <0.001 , eso nos quiere decir que se rechaza H_0 con la hipótesis de que había normalidad multivariada. Lo cual, es un problema importante para el supuesto LDA, entonces procedemos a ocupar un modelo QDA(Quadratic Discriminant Analysis).

Igualdad de Varianzas

Aquí verificamos si la varianza de cada variable es igual entre los grupos, más que nada para verificar o detectar desequilibrios que podrían afectar la estabilidad del modelo, ya desde el punto de vista univariado, dado que solo dos variables cumplen la igualdad de varianzas, el supuesto de homocedasticidad se viola parcialmente.

```
# La varianzas son iguales
leveneTest(Edad~Enfermedad_Cardíaca,data=enfermedad_train_num)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 1  1.9777 0.1601
##      734
```

```
# La varianzas NO son iguales
leveneTest(Presión_Reposo~Enfermedad_Cardíaca,
            data=enfermedad_train_num)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value  Pr(>F)
## group 1  7.2949 0.007075 **
##      734
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# La varianzas NO son iguales
leveneTest(Colesterol~Enfermedad_Cardíaca,data=enfermedad_train_num)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1 107.45 < 2.2e-16 ***
##      734
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# La varianzas son iguales
leveneTest(Frecuencia_Máxima~Enfermedad_Cardíaca,
            data=enfermedad_train_num)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1  0.0747 0.7848
##      734
```

```
# La varianzas NO son iguales
leveneTest(Depresión_ST~Enfermedad_Cardíaca,
            data=enfermedad_train_num)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  1  98.169 < 2.2e-16 ***
##      734
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Matriz de Varianzas y Covarianzas

```
boxM(data=enfermedad_train_num[, -6], grouping=enfermedad_train_num$Enfermedad_Cardíaca)
```

```
##
## Box's M-test for Homogeneity of Covariance Matrices
##
## data:  enfermedad_train_num[, -6]
## Chi-Sq (approx.) = 238.38, df = 15, p-value < 2.2e-16
```

```
# Las matrices de covarianza NO son iguales.
```

Podemos ver que nuestro p-value < 2.2e-16, es decir que rechazamos nuestra hipótesis nula, lo que significa que las matrices de covarianza son significativamente distintas entre grupos. Esta es una violación crítica para LDA, ya que el método asume que todos los grupos tienen la misma estructura de covarianza. La desigualdad genera decisiones de clasificación subóptimas o sesgadas.

Modelo QDA

El modelo QDA (Análisis Discriminante Cuadrático) estima una función discriminante diferente para cada grupo, permitiendo que cada clase tenga su propia matriz de covarianzas y vector de medias. Esto le permite modelar fronteras no lineales entre las clases. Aunque QDA no entrega coeficientes como LDA, podemos identificar la importancia relativa de cada variable comparando las medias de cada grupo. **Cuanto mayor es la diferencia entre medias, mayor es su capacidad para discriminar entre clases.**

```
modelo_qda<-qda(Enfermedad_Cardíaca~, data=enfermedad_train)
modelo_qda$prior # prior probabilities
```

```
## Sin Enfermedad Con Enfermedad
##      0.4375      0.5625
```

```
modelo_qda$scaling # coeficientes / parámetros
```

```

## , , Sin Enfermedad
##
##
##      1      2      3      4
## Edad      0.1066992 0.004890856 -0.02426602 -0.008574099
## SexoM      0.0000000 2.081965928 -0.08009939 -0.083080197
## Dolor_PechoATA 0.0000000 0.000000000 -2.15510680 -1.237001591
## Dolor_PechoNAP 0.0000000 0.000000000 0.00000000 -2.477324196
## Dolor_PechoTA 0.0000000 0.000000000 0.00000000 0.000000000
## Presión_Reposo 0.0000000 0.000000000 0.00000000 0.000000000
## Colesterol    0.0000000 0.000000000 0.00000000 0.000000000
## Glucosa_AyunasSí 0.0000000 0.000000000 0.00000000 0.000000000
## Electrocardiograma_ReposoNormal 0.0000000 0.000000000 0.00000000 0.000000000
## Electrocardiograma_ReposoST 0.0000000 0.000000000 0.00000000 0.000000000
## Frecuencia_Máxima 0.0000000 0.000000000 0.00000000 0.000000000
## Angina_EjercicioSí 0.0000000 0.000000000 0.00000000 0.000000000
## Depresión_ST 0.0000000 0.000000000 0.00000000 0.000000000
## Pendiente_STFlat 0.0000000 0.000000000 0.00000000 0.000000000
## Pendiente_STUp 0.0000000 0.000000000 0.00000000 0.000000000
##
##      5      6      7
## Edad      -0.008556193 0.022423194 0.005574307
## SexoM      0.033407590 0.209759913 0.424924043
## Dolor_PechoATA 0.823211202 0.036342368 -0.141725741
## Dolor_PechoNAP 0.852493833 -0.007280617 0.097008048
## Dolor_PechoTA 4.289411819 0.446900006 0.009604391
## Presión_Reposo 0.000000000 -0.062041485 -0.003223150
## Colesterol    0.000000000 0.000000000 0.015419931
## Glucosa_AyunasSí 0.000000000 0.000000000 0.000000000
## Electrocardiograma_ReposoNormal 0.000000000 0.000000000 0.000000000
## Electrocardiograma_ReposoST 0.000000000 0.000000000 0.000000000
## Frecuencia_Máxima 0.000000000 0.000000000 0.000000000
## Angina_EjercicioSí 0.000000000 0.000000000 0.000000000
## Depresión_ST 0.000000000 0.000000000 0.000000000
## Pendiente_STFlat 0.000000000 0.000000000 0.000000000
## Pendiente_STUp 0.000000000 0.000000000 0.000000000
##
##      8      9      10
## Edad      -0.0255713031 -0.0042979916 -0.011314590
## SexoM      -0.2571445671 0.0953788418 -0.097071324
## Dolor_PechoATA -0.2532684212 0.3700764876 -0.138021554
## Dolor_PechoNAP -0.3683668960 0.1634649854 -0.349869573
## Dolor_PechoTA -0.5038350856 -0.3838717721 -0.493994080
## Presión_Reposo 0.0021714787 -0.0025438904 0.004861328
## Colesterol    -0.0008804994 -0.0003064347 -0.003247872
## Glucosa_AyunasSí 3.2668987392 -0.3503191140 -0.178458430
## Electrocardiograma_ReposoNormal 0.0000000000 -2.1640178899 -1.563799384
## Electrocardiograma_ReposoST 0.0000000000 0.0000000000 -3.553705746
## Frecuencia_Máxima 0.0000000000 0.0000000000 0.0000000000
## Angina_EjercicioSí 0.0000000000 0.0000000000 0.0000000000
## Depresión_ST 0.0000000000 0.0000000000 0.0000000000
## Pendiente_STFlat 0.0000000000 0.0000000000 0.0000000000
## Pendiente_STUp 0.0000000000 0.0000000000 0.0000000000
##
##      11      12      13
## Edad      -0.048880335 -0.0141541722 -0.0185057481
## SexoM      -0.056471410 -0.0088025047 0.0786403138
## Dolor_PechoATA 0.386752283 0.5042786336 0.2709462143
## Dolor_PechoNAP 0.355433363 0.4374806216 -0.0331139945
## Dolor_PechoTA 0.472154471 0.7176127011 -0.6300976515
## Presión_Reposo 0.002116349 -0.0010554278 -0.0033702242
## Colesterol    0.001417860 -0.0001939534 0.0006118138
## Glucosa_AyunasSí 0.116952673 0.0183323836 -0.0725525948
## Electrocardiograma_ReposoNormal -0.413396531 0.0623172535 0.1097038023
## Electrocardiograma_ReposoST -0.770237408 -0.0091379269 0.0774294186
## Frecuencia_Máxima -0.048144588 0.0062170345 -0.0049993545
## Angina_EjercicioSí 0.0000000000 3.1206766056 -0.6198077308
## Depresión_ST 0.0000000000 0.0000000000 1.5407848907
## Pendiente_STFlat 0.0000000000 0.0000000000 0.0000000000
## Pendiente_STUp 0.0000000000 0.0000000000 0.0000000000
##
##      14      15
## Edad      0.0033345983 0.0027663556
## SexoM      -0.1595788077 -0.1775150486
## Dolor_PechoATA -0.2166066653 -0.0537148366

```

```

## Dolor_PechoNAP          -0.0790294061 -0.1756021541
## Dolor_PechoTA           0.0687202355 -0.6246133450
## Presión_Reposo          -0.0052706251  0.0006036685
## Colesterol              -0.0001625777 -0.0007643365
## Glucosa_AyunasSí        -0.1398981554 -0.2362802452
## Electrocardiograma_ReposoNormal -0.2706628297  0.1614100108
## Electrocardiograma_ReposoST -0.3088486208  0.3078900861
## Frecuencia_Máxima       -0.0022067818  0.0002980967
## Angina_EjercicioSí      0.8940549406  0.1262801507
## Depresión_ST            0.6140045481 -0.7013597251
## Pendiente_STFlat        -3.0819961399 -6.5051988234
## Pendiente_STUp          0.0000000000 -7.2747560596
##
## , , Con Enfermedad
##
##              1              2              3
## Edad          -0.1118494 -0.001428042 -0.0007210335
## SexoM          0.0000000 -3.241032533  0.1033660827
## Dolor_PechoATA 0.0000000  0.000000000  4.6606129577
## Dolor_PechoNAP 0.0000000  0.000000000  0.0000000000
## Dolor_PechoTA  0.0000000  0.000000000  0.0000000000
## Presión_Reposo 0.0000000  0.000000000  0.0000000000
## Colesterol     0.0000000  0.000000000  0.0000000000
## Glucosa_AyunasSí 0.0000000  0.000000000  0.0000000000
## Electrocardiograma_ReposoNormal 0.0000000  0.000000000  0.0000000000
## Electrocardiograma_ReposoST 0.0000000  0.000000000  0.0000000000
## Frecuencia_Máxima 0.0000000  0.000000000  0.0000000000
## Angina_EjercicioSí 0.0000000  0.000000000  0.0000000000
## Depresión_ST    0.0000000  0.000000000  0.0000000000
## Pendiente_STFlat 0.0000000  0.000000000  0.0000000000
## Pendiente_STUp  0.0000000  0.000000000  0.0000000000
##              4              5              6
## Edad          -0.01151009  0.003027583  0.02969305
## SexoM          -0.09491222 -0.228060561 -0.47251413
## Dolor_PechoATA 0.43532313  0.238453960  0.37694845
## Dolor_PechoNAP 2.86569150  0.247855130 -0.12599820
## Dolor_PechoTA  0.00000000  5.221734303  0.12078527
## Presión_Reposo 0.00000000  0.000000000 -0.05266940
## Colesterol     0.00000000  0.000000000  0.00000000
## Glucosa_AyunasSí 0.00000000  0.000000000  0.00000000
## Electrocardiograma_ReposoNormal 0.00000000  0.000000000  0.00000000
## Electrocardiograma_ReposoST 0.00000000  0.000000000  0.00000000
## Frecuencia_Máxima 0.00000000  0.000000000  0.00000000
## Angina_EjercicioSí 0.00000000  0.000000000  0.00000000
## Depresión_ST    0.00000000  0.000000000  0.00000000
## Pendiente_STFlat 0.00000000  0.000000000  0.00000000
## Pendiente_STUp  0.00000000  0.000000000  0.00000000
##              7              8              9
## Edad          -0.013809532 -0.006981295 -0.0306448364
## SexoM          -0.405825746  0.077220621 -0.0292857930
## Dolor_PechoATA 0.360694955 -0.067659942 -0.2049241356
## Dolor_PechoNAP 0.017808557  0.080443480 -0.0775001748
## Dolor_PechoTA  0.111091443 -0.160950730 -0.1975845531
## Presión_Reposo 0.011094337 -0.002403282 -0.0028086329
## Colesterol     -0.008244375  0.002681677 -0.0008127475
## Glucosa_AyunasSí 0.000000000  2.260530154 -0.1613377746
## Electrocardiograma_ReposoNormal 0.000000000  0.000000000 -2.1179355283
## Electrocardiograma_ReposoST 0.000000000  0.000000000  0.0000000000
## Frecuencia_Máxima 0.000000000  0.000000000  0.0000000000
## Angina_EjercicioSí 0.000000000  0.000000000  0.0000000000
## Depresión_ST    0.000000000  0.000000000  0.0000000000
## Pendiente_STFlat 0.000000000  0.000000000  0.0000000000
## Pendiente_STUp  0.000000000  0.000000000  0.0000000000
##             10             11             12
## Edad          -0.003024598  0.029250376 -0.003037786
## SexoM          -0.248430081  0.449040200  0.167470793
## Dolor_PechoATA 0.160203917 -0.228838379 -0.646578476
## Dolor_PechoNAP -0.119306085 -0.192743001 -0.371326068
## Dolor_PechoTA  0.222330242 -0.808881592 -0.911465811
## Presión_Reposo -0.001407589  0.005729532  0.007689447
## Colesterol     0.001276083 -0.001649999  0.001318904

```

## Glucosa_AyunasSí	-0.307477421	-0.189317781	-0.292745845
## Electrocardiograma_ReposoNormal	1.559071843	0.615410018	0.128517076
## Electrocardiograma_ReposoST	3.145145241	0.555626381	0.380625177
## Frecuencia_Máxima	0.000000000	0.047435782	-0.010096527
## Angina_EjercicioSí	0.000000000	0.000000000	-2.263451992
## Depresión_ST	0.000000000	0.000000000	0.000000000
## Pendiente_STFlat	0.000000000	0.000000000	0.000000000
## Pendiente_STUp	0.000000000	0.000000000	0.000000000
##	13	14	15
## Edad	-0.017779070	0.010491681	0.003212873
## SexoM	-0.006447987	0.028970570	-0.177683710
## Dolor_PechoATA	0.330929953	-0.041049641	-0.206441613
## Dolor_PechoNAP	0.179273267	-0.340956702	0.094844493
## Dolor_PechoTA	0.096129097	-0.278728776	0.157416914
## Presión_Reposo	-0.004791482	-0.007170629	-0.004032947
## Colesterol	-0.001391978	-0.001519791	-0.000233797
## Glucosa_AyunasSí	0.041326473	-0.059634882	0.233251663
## Electrocardiograma_ReposoNormal	-0.034034735	-0.335676303	0.067255296
## Electrocardiograma_ReposoST	-0.108518166	-0.047399472	-0.011868200
## Frecuencia_Máxima	-0.005600880	0.010245101	-0.007125329
## Angina_EjercicioSí	-0.485579994	-0.152048520	0.262349630
## Depresión_ST	0.929887557	0.124529099	0.304708103
## Pendiente_STFlat	0.000000000	2.477185380	2.654463652
## Pendiente_STUp	0.000000000	0.000000000	4.431571443

```
predicciones<-predict(object=modelo_qda,newdata=enfermedad_test)
predicciones$posterior
```

##	Sin Enfermedad	Con Enfermedad
## 1	9.548246e-08	9.999999e-01
## 2	6.843071e-10	1.000000e+00
## 4	9.994513e-01	5.486752e-04
## 7	2.320611e-02	9.767939e-01
## 9	9.999991e-01	9.381664e-07
## 25	7.114465e-01	2.885535e-01
## 30	1.107749e-09	1.000000e+00
## 32	9.999992e-01	7.789477e-07
## 33	8.222642e-03	9.917774e-01
## 37	5.092263e-04	9.994908e-01
## 49	1.568113e-03	9.984319e-01
## 54	1.669458e-01	8.330542e-01
## 65	9.999951e-01	4.934909e-06
## 71	9.915556e-01	8.444363e-03
## 78	1.195292e-01	8.804708e-01
## 81	2.498681e-07	9.999998e-01
## 84	1.722396e-06	9.999983e-01
## 86	8.339002e-01	1.660998e-01
## 96	9.489294e-01	5.107061e-02
## 98	4.749018e-04	9.995251e-01
## 102	8.760647e-04	9.991239e-01
## 103	9.999999e-01	8.133707e-08
## 121	1.014074e-01	8.985926e-01
## 128	9.999639e-01	3.614026e-05
## 141	9.999999e-01	6.449729e-08
## 146	3.132811e-02	9.686719e-01
## 147	1.616174e-02	9.838383e-01
## 148	3.606271e-04	9.996394e-01
## 150	5.614337e-01	4.385663e-01
## 169	9.993006e-01	6.994292e-04
## 176	7.852329e-02	9.214767e-01
## 178	1.903094e-05	9.999810e-01
## 182	9.978842e-01	2.115829e-03
## 183	4.060626e-04	9.995939e-01
## 187	7.491627e-06	9.999925e-01
## 189	9.999969e-01	3.074384e-06
## 193	1.139806e-02	9.886019e-01
## 195	9.999926e-01	7.423926e-06
## 200	9.999998e-01	1.981013e-07
## 208	8.129812e-04	9.991870e-01
## 213	9.999636e-01	3.642399e-05
## 217	5.701993e-03	9.942980e-01
## 227	9.772404e-05	9.999023e-01
## 244	9.995142e-01	4.857729e-04
## 249	9.759284e-01	2.407158e-02
## 253	2.704886e-10	1.000000e+00
## 254	2.061623e-09	1.000000e+00
## 255	9.816585e-01	1.834153e-02
## 257	2.200457e-02	9.779954e-01
## 262	1.043055e-03	9.989569e-01
## 263	2.718525e-07	9.999997e-01
## 264	9.998692e-01	1.308063e-04
## 270	2.360020e-01	7.639980e-01
## 275	8.691452e-03	9.913085e-01
## 277	1.323143e-01	8.676857e-01
## 283	9.993629e-01	6.370663e-04
## 288	1.577276e-02	9.842272e-01
## 294	8.581973e-04	9.991418e-01
## 296	8.000948e-02	9.199905e-01
## 299	1.000000e+00	4.699664e-08
## 302	2.278402e-02	9.772160e-01
## 316	3.350711e-10	1.000000e+00
## 321	5.959191e-02	9.404081e-01
## 324	7.004106e-07	9.999993e-01
## 325	2.270455e-03	9.977295e-01
## 330	9.999907e-01	9.269205e-06
## 334	2.893724e-04	9.997106e-01
## 342	1.906132e-04	9.998094e-01
## 348	9.999834e-01	1.656998e-05


```

## 355 4.504113e-05 9.999550e-01
## 357 2.487803e-13 1.000000e+00
## 359 1.751229e-02 9.824877e-01
## 370 4.133490e-06 9.999959e-01
## 374 5.959813e-07 9.999994e-01
## 377 9.999727e-01 2.732453e-05
## 382 1.792787e-02 9.820721e-01
## 390 9.999995e-01 4.514940e-07
## 391 9.960923e-01 3.907662e-03
## 392 9.999991e-01 9.480877e-07
## 401 5.712108e-01 4.287892e-01
## 403 9.999965e-01 3.522122e-06
## 405 1.777110e-03 9.982229e-01
## 408 1.114819e-06 9.999989e-01
## 426 9.999606e-01 3.939650e-05
## 432 1.000000e+00 2.087222e-08
## 435 9.995042e-01 4.957850e-04
## 444 1.279669e-03 9.987203e-01
## 450 9.889130e-01 1.108703e-02
## 451 9.962583e-01 3.741743e-03
## 452 5.057197e-05 9.999494e-01
## 456 9.990778e-01 9.221632e-04
## 460 9.999960e-01 3.997781e-06
## 465 9.999988e-01 1.242287e-06
## 474 2.790343e-02 9.720966e-01
## 475 1.419533e-01 8.580467e-01
## 483 9.998647e-01 1.352888e-04
## 495 9.886960e-01 1.130404e-02
## 497 9.720065e-01 2.799346e-02
## 507 9.970209e-01 2.979093e-03
## 514 2.639469e-02 9.736053e-01
## 527 9.999996e-01 3.537817e-07
## 529 9.999940e-01 6.031722e-06
## 531 5.778604e-01 4.221396e-01
## 533 9.526542e-01 4.734578e-02
## 534 3.865020e-01 6.134980e-01
## 538 9.999992e-01 7.708045e-07
## 545 2.094283e-09 1.000000e+00
## 551 1.618060e-02 9.838194e-01
## 552 1.440868e-04 9.998559e-01
## 564 9.999994e-01 6.376235e-07
## 570 2.920655e-04 9.997079e-01
## 571 1.184224e-02 9.881578e-01
## 573 9.859103e-01 1.408971e-02
## 575 9.778981e-01 2.210186e-02
## 579 9.999986e-01 1.360757e-06
## 583 2.796877e-03 9.972031e-01
## 589 9.782115e-01 2.178846e-02
## 596 2.504586e-02 9.749541e-01
## 599 1.166225e-03 9.988338e-01
## 604 9.869009e-01 1.309907e-02
## 612 9.607698e-01 3.923022e-02
## 613 9.833181e-01 1.668194e-02
## 616 6.352502e-03 9.936475e-01
## 620 9.748855e-01 2.511455e-02
## 624 8.575085e-01 1.424915e-01
## 625 7.736974e-05 9.999226e-01
## 630 4.103066e-05 9.999590e-01
## 641 9.344220e-11 1.000000e+00
## 643 9.999855e-01 1.452548e-05
## 665 9.995936e-01 4.064239e-04
## 666 1.230044e-01 8.769956e-01
## 671 1.000000e+00 1.917009e-08
## 675 9.999343e-01 6.570844e-05
## 678 3.130298e-02 9.686970e-01
## 680 1.212251e-03 9.987877e-01
## 687 2.436404e-03 9.975636e-01
## 689 5.504986e-03 9.944950e-01
## 699 8.660774e-05 9.999134e-01
## 700 5.226118e-03 9.947739e-01
## 702 3.875071e-09 1.000000e+00

```

```
## 705 6.116230e-03 9.938838e-01
## 711 4.510544e-01 5.489456e-01
## 721 3.036418e-04 9.996964e-01
## 722 9.828534e-01 1.714660e-02
## 724 2.829794e-04 9.997170e-01
## 740 9.811110e-01 1.888896e-02
## 746 5.594611e-01 4.405389e-01
## 747 6.185015e-03 9.938150e-01
## 757 5.963808e-02 9.403619e-01
## 760 3.136956e-01 6.863044e-01
## 762 8.627369e-03 9.913726e-01
## 767 4.990290e-11 1.000000e+00
## 770 9.979180e-01 2.082020e-03
## 771 7.700024e-03 9.923000e-01
## 776 1.311211e-07 9.999999e-01
## 779 9.999883e-01 1.170702e-05
## 784 9.999988e-01 1.230915e-06
## 804 5.646149e-01 4.353851e-01
## 805 4.349057e-04 9.995651e-01
## 806 5.822333e-04 9.994178e-01
## 807 1.329685e-03 9.986703e-01
## 813 6.041719e-01 3.958281e-01
## 817 9.169612e-01 8.303876e-02
## 820 1.654749e-04 9.998345e-01
## 821 1.546873e-02 9.845313e-01
## 822 9.994136e-01 5.864491e-04
## 831 1.952990e-02 9.804701e-01
## 833 8.727793e-01 1.272207e-01
## 851 3.724506e-05 9.999628e-01
## 864 9.900639e-01 9.936128e-03
## 866 9.989506e-01 1.049400e-03
## 876 9.999995e-01 5.323193e-07
## 879 8.055444e-01 1.944556e-01
## 889 5.514978e-04 9.994485e-01
## 891 1.000000e+00 4.192221e-08
## 893 9.989715e-01 1.028549e-03
## 895 9.999973e-01 2.710543e-06
## 911 4.562707e-06 9.999954e-01
## 912 9.753094e-01 2.469064e-02
## 914 8.185800e-13 1.000000e+00
## 917 1.991696e-05 9.999801e-01
## 918 3.135748e-03 9.968643e-01
```

```
clase_predicha<-predicciones$class
clase_predicha
```

```
## [1] Con Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad
## [6] Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad
## [11] Con Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad
## [16] Con Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad
## [21] Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad
## [26] Con Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad
## [31] Con Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad
## [36] Sin Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad
## [41] Sin Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad
## [46] Con Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad
## [51] Con Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad
## [56] Sin Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad
## [61] Con Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad
## [66] Sin Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
## [71] Con Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad
## [76] Con Enfermedad Sin Enfermedad Sin Enfermedad Sin Enfermedad Sin Enfermedad
## [81] Sin Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad
## [86] Sin Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad
## [91] Sin Enfermedad Sin Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad
## [96] Sin Enfermedad Sin Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad
## [101] Sin Enfermedad Sin Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad
## [106] Sin Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad
## [111] Con Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad Sin Enfermedad
## [116] Con Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad
## [121] Sin Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad
## [126] Con Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad Sin Enfermedad
## [131] Con Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad
## [136] Con Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad
## [141] Con Enfermedad Con Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
## [146] Sin Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad Con Enfermedad
## [151] Con Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad
## [156] Sin Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad
## [161] Con Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad Con Enfermedad
## [166] Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad
## [171] Sin Enfermedad Sin Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad
## [176] Sin Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
## [181] Con Enfermedad Con Enfermedad
## Levels: Sin Enfermedad Con Enfermedad
```

- Frecuencia_Máxima: Es la variable más discriminante. Los pacientes sin enfermedad tienden a alcanzar frecuencias cardíacas máximas más altas.
- Depresión_ST: Segundo indicador más importante; los pacientes con enfermedad tienen valores mucho mayores.
- Colesterol y Edad también presentan diferencias marcadas, siendo útiles para la separación.
- Variables como Angina_EjercicioSí y Pendiente_STUp también aportan información valiosa.
- Algunas variables categóricas (como Dolor_Pecho en sus variantes) también aportan discriminación, aunque en menor grado.

Matriz de Confusión

```
dato_real<-enfermedad_test$Enfermedad_Cardíaca
dato_predicho<-factor(clase_predicha,levels=c("Sin Enfermedad","Con Enfermedad"))

# Matriz de confusión y métricas
conf_matrix <- confusionMatrix(data=dato_predicho,reference=dato_real,
                                positive="Con Enfermedad")

print(conf_matrix$table)
```

```
##           Reference
## Prediction Sin Enfermedad Con Enfermedad
## Sin Enfermedad           72           10
## Con Enfermedad           16           84
```

```
# Métricas individuales
accuracy <- conf_matrix$overall["Accuracy"] # Proporción total de predicciones correctas.
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8571429
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.84
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.893617
```

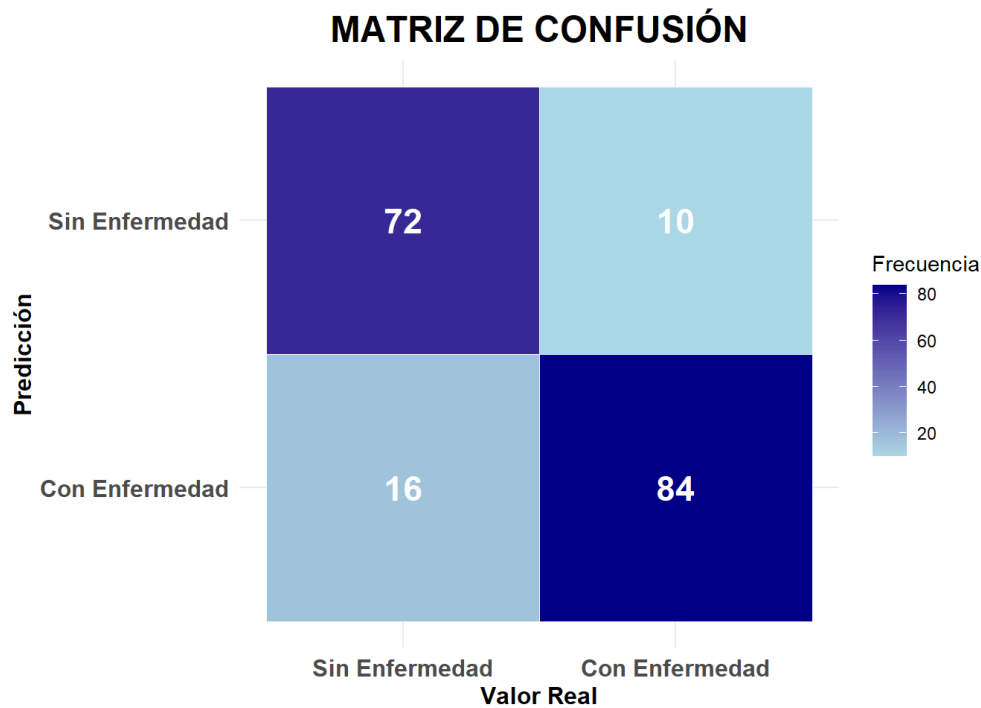
```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.8659794
```

```
cm_table <- as.data.frame(conf_matrix$table)

# Invertir el orden de los niveles del eje Y
cm_table$Prediction <- factor(cm_table$Prediction, levels = rev(levels(cm_table$Prediction)))

ggplot(cm_table, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), color = "white", size = 6, fontface = "bold") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name = "Frecuencia") +
  labs(
    title = "MATRIZ DE CONFUSIÓN",
    x = "Valor Real",
    y = "Predicción"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 12, face = "bold")
  )
```



- Accuracy (Exactitud): El 85.7% de las predicciones fueron correctas.
- Precision (Precisión): El 84% de las veces que se predijo "Con Enfermedad", realmente era cierto.
- Recall (Sensibilidad): El modelo identifica correctamente el 89.4% de los pacientes que realmente tienen enfermedad.
- F1 Score: Es el promedio armónico entre precisión y recall. Una métrica balanceada, útil si hay desbalance de clases.

Chequemos los dos posibles errores, en este caso en FP, el cual tenemos 16, se predijo enfermedad en pacientes sanos, esto puede generar ansiedad o intervenciones médicas innecesarias. Ahora, el otro error, los FN, que tenemos 10, se predijo "sin enfermedad" en pacientes enfermos, esto es crítico, ya que el paciente enfermo no recibe tratamiento a tiempo. En este sentido nos interesa la sensibilidad, ya que es crítico en contextos médicos para no pasar por alto pacientes enfermos

ÁRBOL DE DECISIÓN

El planteamiento para esta técnica será crear 5 árboles de decisión distintos donde cada uno será producto de una modificación en la función `rpart`, al final se elegirá el que arroje la mejor métrica del Recall o Sensibilidad. Lo que se va a realizar es:

- Índice de Gini
- Profundidad
- Ganancia de Información
- Validación Cruzada
- Poda del Árbol

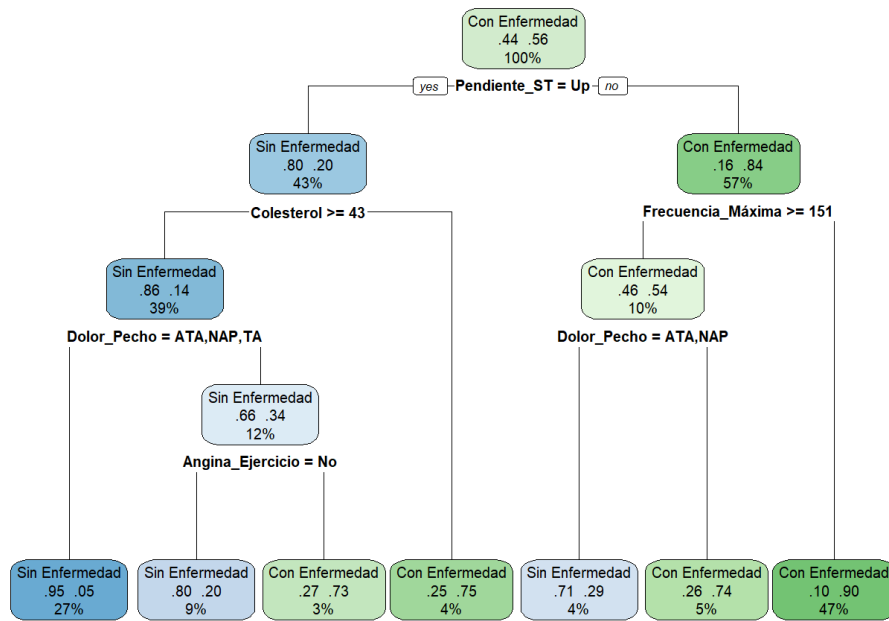
Índice de Gini

Entrenamiento del árbol con el índice de Gini, el índice de Gini es una medida de impureza que indica la probabilidad de que una instancia sea clasificada incorrectamente.

```
# Entrenamiento del árbol con índice Gini
modelo <- rpart(Enfermedad_Cardíaca ~ ., data = enfermedad_train, method = "class", parms = list(split = "gini"))
```

Visualizamos el árbol

```
rpart.plot(modelo, type = 2, extra = 104)
```



Realizamos nuestras predicciones

```
y_pred <- predict(modelo, newdata = enfermedad_test, type = "class")
y_pred
```

##	1	2	4	7	9
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	25	30	32	33	37
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	49	54	65	71	78
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	81	84	86	96	98
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	102	103	121	128	141
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	146	147	148	150	169
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	176	178	182	183	187
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	189	193	195	200	208
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	213	217	227	244	249
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	253	254	255	257	262
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	263	264	270	275	277
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	283	288	294	296	299
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	302	316	321	324	325
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	330	334	342	348	355
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	357	359	370	374	377
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	382	390	391	392	401
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	403	405	408	426	432
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	435	444	450	451	452
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	456	460	465	474	475
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	483	495	497	507	514
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	527	529	531	533	534
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	538	545	551	552	564
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	570	571	573	575	579
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	583	589	596	599	604
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	612	613	616	620	624
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	625	630	641	643	665
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	666	671	675	678	680
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	687	689	699	700	702
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	705	711	721	722	724
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	740	746	747	757	760
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	762	767	770	771	776
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	779	784	804	805	806
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	807	813	817	820	821
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	822	831	833	851	864
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	866	876	879	889	891
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad

```
##           893           895           911           912           914
## Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
##           917           918
## Con Enfermedad Con Enfermedad
## Levels: Sin Enfermedad Con Enfermedad
```

Generamos la matriz de confusión

```
y_real <- enfermedad_test$Enfermedad_Cardíaca

conf_matrix <- confusionMatrix(data = y_pred, reference = y_real, positive = "Con Enfermedad")
print(conf_matrix$table)
```

```
##           Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad           62           7
## Con Enfermedad          26          87
```

Extraemos las métricas

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8186813
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.7699115
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.9255319
```

```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.8405797
```

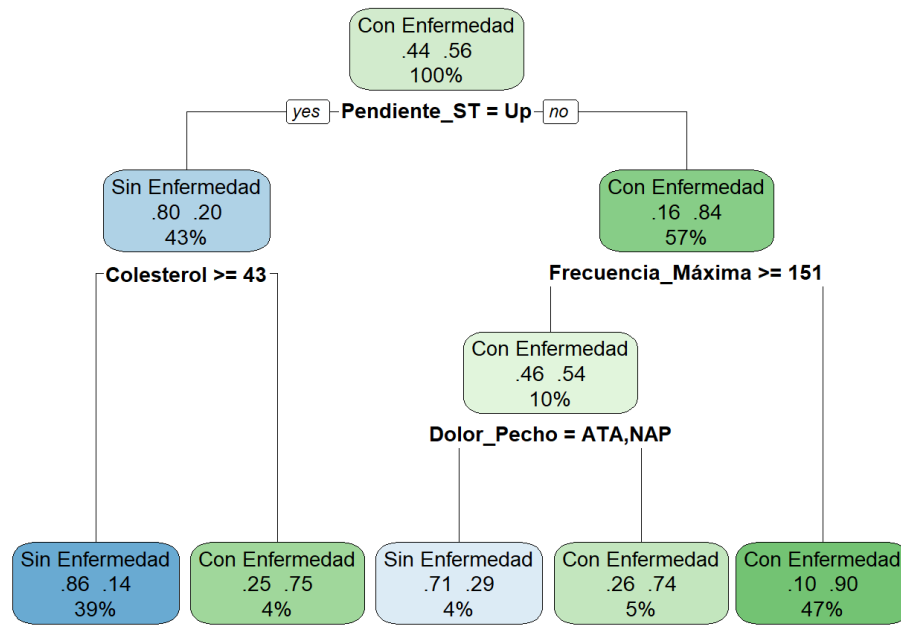
Máximo de Niveles

Se trabajará con el mejor árbol que cumpla tener 3 o menos niveles

```
modelo2 <- rpart(Enfermedad_Cardíaca ~ ., data = enfermedad_train, method = "class", parms = list(split = "gini"), control =
rpart.control(maxdepth = 3))
```

Visualizamos el árbol

```
rpart.plot(modelo2, type = 2, extra = 104)
```

Realizamos nuestras predicciones

```
y_pred <- predict(modelo2, newdata = enfermedad_test, type = "class")
y_pred
```

##	1	2	4	7	9
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	25	30	32	33	37
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	49	54	65	71	78
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	81	84	86	96	98
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	102	103	121	128	141
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	146	147	148	150	169
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	176	178	182	183	187
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	189	193	195	200	208
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	213	217	227	244	249
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	253	254	255	257	262
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	263	264	270	275	277
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	283	288	294	296	299
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	302	316	321	324	325
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	330	334	342	348	355
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	357	359	370	374	377
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	382	390	391	392	401
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	403	405	408	426	432
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	435	444	450	451	452
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	456	460	465	474	475
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	483	495	497	507	514
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	527	529	531	533	534
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	538	545	551	552	564
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	570	571	573	575	579
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	583	589	596	599	604
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	612	613	616	620	624
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	625	630	641	643	665
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	666	671	675	678	680
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	687	689	699	700	702
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	705	711	721	722	724
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	740	746	747	757	760
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	762	767	770	771	776
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	779	784	804	805	806
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	807	813	817	820	821
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	822	831	833	851	864
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	866	876	879	889	891
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad

```
##           893           895           911           912           914
## Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
##           917           918
## Con Enfermedad Con Enfermedad
## Levels: Sin Enfermedad Con Enfermedad
```

Generamos la matriz de confusión

```
y_real <- enfermedad_test$Enfermedad_Cardíaca

conf_matrix <- confusionMatrix(data = y_pred, reference = y_real, positive = "Con Enfermedad")
print(conf_matrix$table)
```

```
##           Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad           64           11
## Con Enfermedad          24           83
```

Extraemos las métricas

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8076923
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.7757009
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.8829787
```

```
cat("F1 Score:", f1, "\n")
```

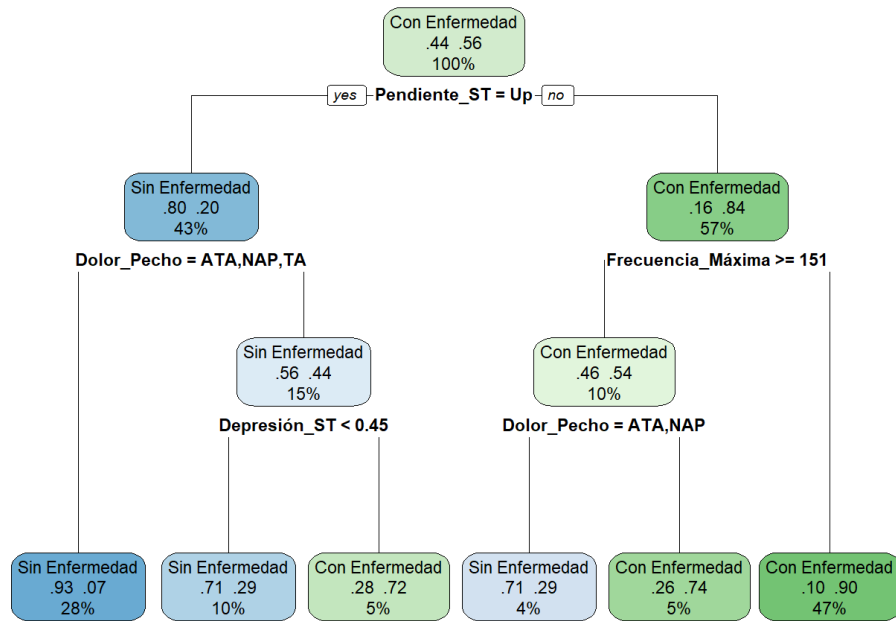
```
## F1 Score: 0.8258706
```

Ganancia de Información

```
modelo3 <- rpart(Enfermedad_Cardíaca ~ ., data = enfermedad_train, method = "class", parms = list(split = "information"))
```

Visualizamos el árbol

```
rpart.plot(modelo3, type = 2, extra = 104)
```



Realizamos nuestras predicciones

```
y_pred <- predict(modelo3, newdata = enfermedad_test, type = "class")
y_pred
```

##	1	2	4	7	9
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	25	30	32	33	37
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	49	54	65	71	78
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	81	84	86	96	98
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	102	103	121	128	141
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	146	147	148	150	169
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	176	178	182	183	187
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	189	193	195	200	208
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	213	217	227	244	249
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	253	254	255	257	262
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	263	264	270	275	277
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	283	288	294	296	299
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	302	316	321	324	325
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	330	334	342	348	355
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	357	359	370	374	377
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	382	390	391	392	401
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	403	405	408	426	432
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	435	444	450	451	452
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	456	460	465	474	475
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	483	495	497	507	514
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	527	529	531	533	534
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	538	545	551	552	564
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	570	571	573	575	579
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	583	589	596	599	604
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	612	613	616	620	624
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	625	630	641	643	665
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	666	671	675	678	680
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	687	689	699	700	702
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	705	711	721	722	724
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	740	746	747	757	760
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	762	767	770	771	776
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	779	784	804	805	806
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	807	813	817	820	821
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	822	831	833	851	864
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	866	876	879	889	891
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad

```
##           893           895           911           912           914
## Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
##           917           918
## Con Enfermedad Con Enfermedad
## Levels: Sin Enfermedad Con Enfermedad
```

Generamos la matriz de confusión

```
y_real <- enfermedad_test$Enfermedad_Cardíaca

conf_matrix <- confusionMatrix(data = y_pred, reference = y_real, positive = "Con Enfermedad")
print(conf_matrix$table)
```

```
##           Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad           69           12
## Con Enfermedad          19           82
```

Extraemos las métricas

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8296703
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.8118812
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.8723404
```

```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.8410256
```

Validación Cruzada

Ajustamos el modelo con un cp bajo de 0.001 y 10 folds (divisiones), activamos validación cruzada.

```
modelo4 <- rpart(Enfermedad_Cardíaca ~ ., data = enfermedad_train, method = "class", control = rpart.control(cp = 0.001, xval = 10))
```

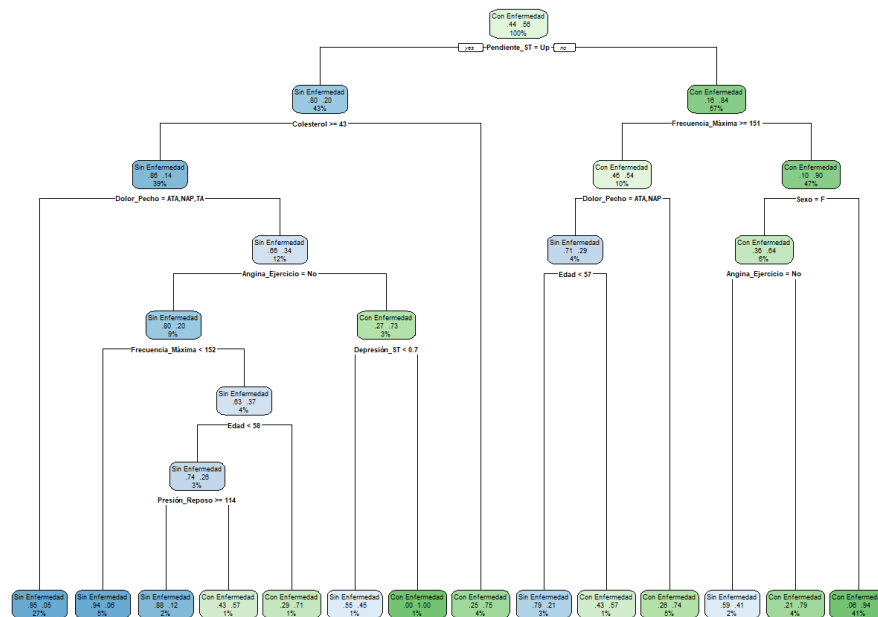
Revisamos la tabla de complejidad del modelo.

```
printcp(modelo4)
```

```
##
## Classification tree:
## rpart(formula = Enfermedad_Cardíaca ~ ., data = enfermedad_train,
##       method = "class", control = rpart.control(cp = 0.001, xval = 10))
##
## Variables actually used in tree construction:
## [1] Angina_Ejercicio Colesterol Depresión_ST Dolor_Pecho
## [5] Edad Frecuencia_Máxima Pendiente_ST Presión_Reposo
## [9] Sexo
##
## Root node error: 322/736 = 0.4375
##
## n= 736
##
##      CP nsplit rel error  xerror  xstd
## 1 0.5962733      0  1.00000 1.00000 0.041796
## 2 0.0496894      1  0.40373 0.40373 0.032130
## 3 0.0201863      2  0.35404 0.37888 0.031331
## 4 0.0155280      4  0.31366 0.34783 0.030263
## 5 0.0046584      6  0.28261 0.33230 0.029698
## 6 0.0031056     10  0.26398 0.33851 0.029926
## 7 0.0010000     13  0.25466 0.34472 0.030151
```

Visualizamos el árbol

```
rpart.plot(modelo4, type = 2, extra = 104)
```



Realizamos nuestras predicciones

```
y_pred <- predict(modelo4, newdata = enfermedad_test, type = "class")
y_pred
```

##	1	2	4	7	9
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	25	30	32	33	37
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	49	54	65	71	78
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	81	84	86	96	98
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	102	103	121	128	141
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	146	147	148	150	169
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	176	178	182	183	187
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	189	193	195	200	208
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	213	217	227	244	249
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	253	254	255	257	262
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	263	264	270	275	277
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	283	288	294	296	299
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	302	316	321	324	325
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	330	334	342	348	355
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	357	359	370	374	377
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	382	390	391	392	401
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	403	405	408	426	432
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	435	444	450	451	452
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	456	460	465	474	475
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	483	495	497	507	514
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	527	529	531	533	534
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	538	545	551	552	564
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	570	571	573	575	579
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	583	589	596	599	604
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	612	613	616	620	624
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	625	630	641	643	665
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	666	671	675	678	680
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	687	689	699	700	702
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	705	711	721	722	724
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	740	746	747	757	760
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	762	767	770	771	776
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	779	784	804	805	806
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	807	813	817	820	821
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	822	831	833	851	864
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	866	876	879	889	891
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad


```
##           893           895           911           912           914
## Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
##           917           918
## Con Enfermedad Con Enfermedad
## Levels: Sin Enfermedad Con Enfermedad
```

Generamos la matriz de confusión

```
y_real <- enfermedad_test$Enfermedad_Cardíaca

conf_matrix <- confusionMatrix(data = y_pred, reference = y_real, positive = "Con Enfermedad")
print(conf_matrix$table)
```

```
##           Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad           68           8
## Con Enfermedad          20          86
```

Extraemos las métricas

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8461538
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.8113208
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.9148936
```

```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.86
```

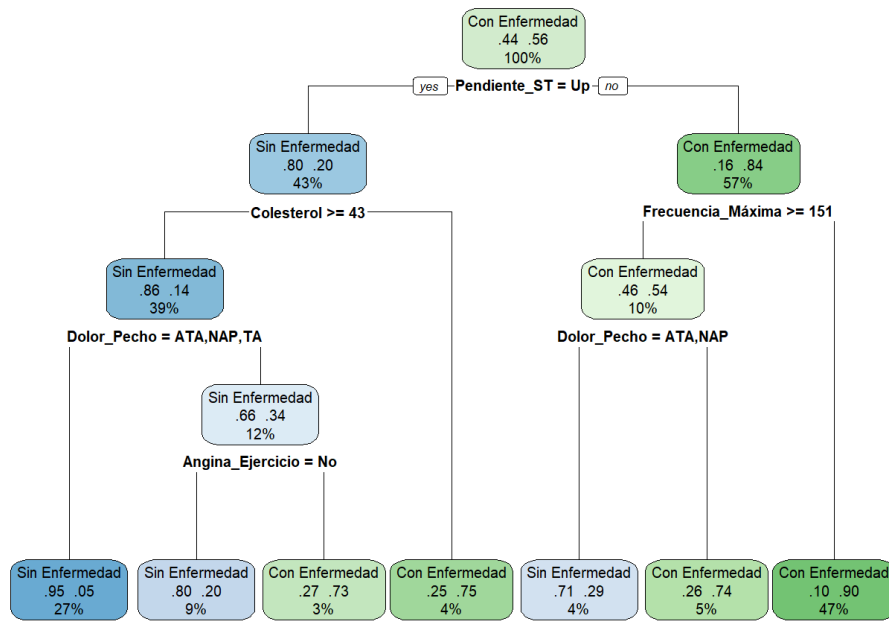
Árbol Podado

Podamos el árbol al mejor cp

```
mejor_cp <- modelo4$cptable[which.min(modelo4$cptable[, "xerror"]), "CP"]
modelo4_podado <- prune(modelo4, cp = mejor_cp)
```

Visualizamos el árbol después de la poda

```
rpert.plot(modelo4_podado, type = 2, extra = 104)
```



Realizamos nuestras predicciones

```
y_pred <- predict(modelo4_podado, newdata = enfermedad_test, type = "class")
y_pred
```

##	1	2	4	7	9
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	25	30	32	33	37
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	49	54	65	71	78
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	81	84	86	96	98
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	102	103	121	128	141
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	146	147	148	150	169
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	176	178	182	183	187
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	189	193	195	200	208
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	213	217	227	244	249
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	253	254	255	257	262
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	263	264	270	275	277
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	283	288	294	296	299
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	302	316	321	324	325
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	330	334	342	348	355
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	357	359	370	374	377
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	382	390	391	392	401
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	403	405	408	426	432
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	435	444	450	451	452
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	456	460	465	474	475
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	483	495	497	507	514
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	527	529	531	533	534
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	538	545	551	552	564
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	570	571	573	575	579
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	583	589	596	599	604
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	612	613	616	620	624
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	625	630	641	643	665
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	666	671	675	678	680
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	687	689	699	700	702
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	705	711	721	722	724
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	740	746	747	757	760
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	762	767	770	771	776
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	779	784	804	805	806
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	807	813	817	820	821
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	822	831	833	851	864
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	866	876	879	889	891
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad

```
##           893           895           911           912           914
## Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
##           917           918
## Con Enfermedad Con Enfermedad
## Levels: Sin Enfermedad Con Enfermedad
```

Generamos la matriz de confusión

```
y_real <- enfermedad_test$Enfermedad_Cardíaca

conf_matrix <- confusionMatrix(data = y_pred, reference = y_real, positive = "Con Enfermedad")
print(conf_matrix$table)
```

```
##           Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad           62           7
## Con Enfermedad          26          87
```

Extraemos las métricas

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8186813
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.7699115
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.9255319
```

```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.8405797
```

Elección del Mejor Árbol

Métrica	Índice de Gini	Máximo de Niveles	Ganancia Información	Validación Cruzada	Poda
Verdaderos Negativos (TN)	62	64	69	68	62
Falsos Positivos (FP)	7	11	12	8	7
Falsos Negativos (FN)	26	24	19	20	26
Verdaderos Positivos (TP)	87	83	82	86	87
Exactitud (Accuracy)	81.86%	80.76%	82.96%	84.61%	81.86
Precisión (Precision)	76.99%	77.57%	81.18%	81.13%	76.99
Sensibilidad (Recall)	92.55%	88.29%	87.23%	91.48%	92.55
F1 Score	84.05%	82.85%	84.1%	86%	84.05

Elección del mejor modelo.

Validación Cruzada:

- Exactitud (84.61%): La más alta entre todos los modelos.
- Precisión (81.13%): Segunda más alta, cercana al modelo de "Ganancia Información".

- Sensibilidad (91.48%): Muy alta, solo superada por los modelos de "Índice de Gini" y "Podado".
- F1 Score (86%): El más alto, lo que indica un buen equilibrio entre precisión y sensibilidad.

Ganancia Información:

- Buen desempeño en exactitud (82.96%) y F1 Score (84.1%), pero con menor sensibilidad (87.23%) comparado con la validación cruzada.

Índice de Gini y Podado:

- Alta sensibilidad (92.55%), pero menor precisión (76.99%) y exactitud (81.86%). Esto sugiere que pueden estar sobreajustados o favorecer falsos positivos.

Se puede concluir que el modelo de **Árbol Podado** es el mejor debido al valor más alto que maneja en el Recall de 92.55% que va alineado con los objetivos del análisis.

Mejor Árbol

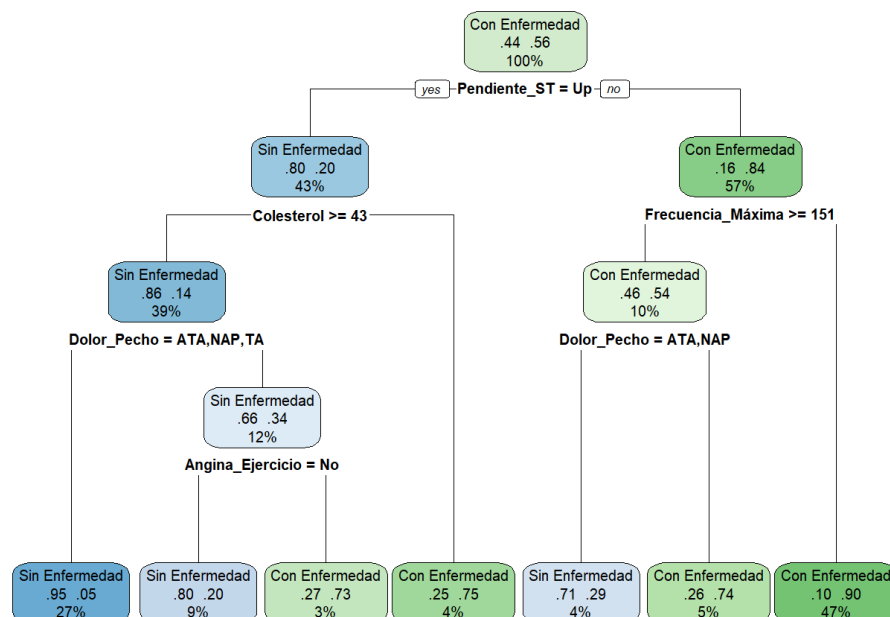
Podamos el árbol al mejor cp

```
mejor_cp <- modelo4$cptable[which.min(modelo4$cptable[, "xerror"]), "CP"]
modelo4_podado <- prune(modelo4, cp = mejor_cp)
mejor_cp
```

```
## [1] 0.004658385
```

Visualizamos el árbol después de la poda

```
rpert.plot(modelo4_podado, type = 2, extra = 104)
```



- cp: Parámetro de complejidad; su función es determinar que tan buena debe ser una división para que el algoritmo decida realizarla
- nsplit: Divisiones en el árbol, nsplit=0 es el nodo raíz
- rel error: Error relativo respecto al error del nodo raíz
- xerror: Desviación estándar del error de validación cruzada. Ayuda a evaluar la estabilidad del modelo.
- El mejor cp es el que tiene el xerror más bajo, en este caso 0.32919, un cp de 0.0046584 y 6 divisiones

El árbol está dividido en dos ramas principales según la variable Pendiente_ST, que son la subida o bajada del electrocardiograma, de ahí, cada rama se subdivide según otras variables médicas, como el colesterol, dolor de pecho, anginas y frecuencia.

La predicción del nodo terminal es si el paciente está enfermo o no, el porcentaje indica los casos en ese nodo que tienen alguna enfermedad cardíaca y también se presenta la proporción total de pacientes que caen en ese nodo

Ejemplo de interpretación: Supongamos que un paciente tiene:

- Pendiente_ST = Up
- Colesterol ≥ 43
- Dolor_Pecho = ASY
- Angina_Ejercicio = No

→ Este paciente terminaría en un nodo hoja que predice "Sin Enfermedad" con una precisión del 80% y representa al 9% de los casos.

Notemos que Pendiente_ST es la variable más importante (raíz del árbol): indica su alta relevancia en la predicción.

En la rama derecha (Down, Flat): La Frecuencia Máxima y el Dolor de Pecho tienen un papel relevante en la predicción.

Realizamos nuestras predicciones

```
y_pred <- predict(modelo4_podado, newdata = enfermedad_test, type = "class")  
y_pred
```

##	1	2	4	7	9
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad
##	25	30	32	33	37
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	49	54	65	71	78
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	81	84	86	96	98
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	102	103	121	128	141
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	146	147	148	150	169
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	176	178	182	183	187
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	189	193	195	200	208
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	213	217	227	244	249
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	253	254	255	257	262
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	263	264	270	275	277
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	283	288	294	296	299
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	302	316	321	324	325
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	330	334	342	348	355
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	357	359	370	374	377
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	382	390	391	392	401
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	403	405	408	426	432
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	435	444	450	451	452
##	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	456	460	465	474	475
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	483	495	497	507	514
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	527	529	531	533	534
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad
##	538	545	551	552	564
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	570	571	573	575	579
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad
##	583	589	596	599	604
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	612	613	616	620	624
##	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad
##	625	630	641	643	665
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Sin Enfermedad
##	666	671	675	678	680
##	Con Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	687	689	699	700	702
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	705	711	721	722	724
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	740	746	747	757	760
##	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	762	767	770	771	776
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	779	784	804	805	806
##	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad	Sin Enfermedad
##	807	813	817	820	821
##	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad	Con Enfermedad
##	822	831	833	851	864
##	Con Enfermedad	Con Enfermedad	Sin Enfermedad	Con Enfermedad	Con Enfermedad
##	866	876	879	889	891
##	Sin Enfermedad	Sin Enfermedad	Sin Enfermedad	Con Enfermedad	Sin Enfermedad

```
##           893           895           911           912           914
## Con Enfermedad Sin Enfermedad Con Enfermedad Sin Enfermedad Con Enfermedad
##           917           918
## Con Enfermedad Con Enfermedad
## Levels: Sin Enfermedad Con Enfermedad
```

Generamos la matriz de confusión

```
y_real <- enfermedad_test$Enfermedad_Cardíaca

conf_matrix <- confusionMatrix(data = y_pred, reference = y_real, positive = "Con Enfermedad")
print(conf_matrix$table)
```

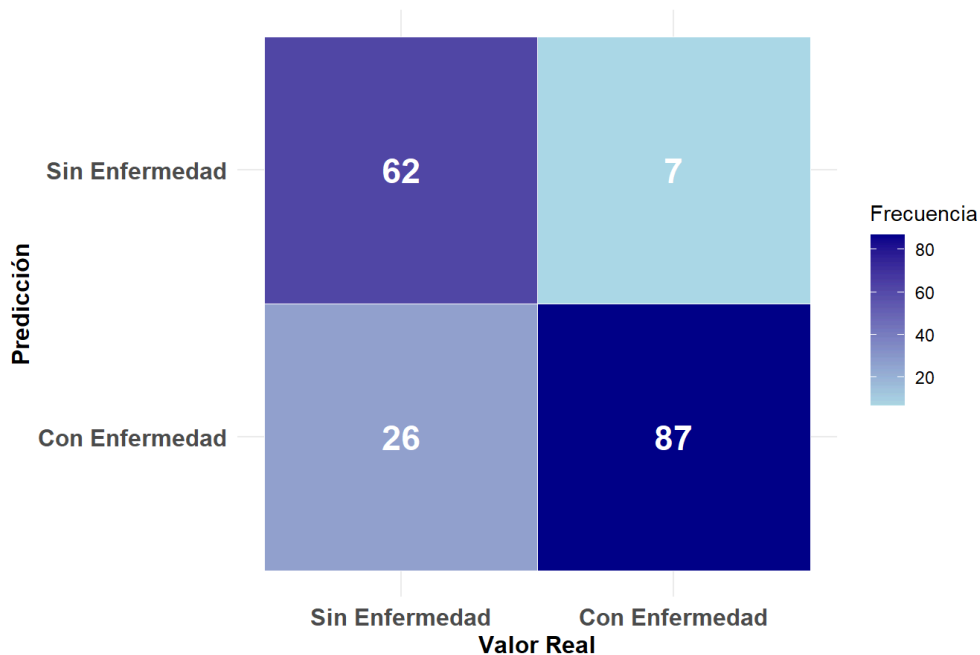
```
##           Reference
## Prediction   Sin Enfermedad Con Enfermedad
## Sin Enfermedad      62         7
## Con Enfermedad     26        87
```

```
cm_table <- as.data.frame(conf_matrix$table)

# Invertir el orden de los niveles del eje Y
cm_table$Prediction <- factor(cm_table$Prediction, levels = rev(levels(cm_table$Prediction)))

ggplot(cm_table, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), color = "white", size = 6, fontface = "bold") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name = "Frecuencia") +
  labs(
    title = "MATRIZ DE CONFUSIÓN",
    x = "Valor Real",
    y = "Predicción"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 12, face = "bold")
  )
```

MATRIZ DE CONFUSIÓN



- Verdaderos negativos (TN = 62): El modelo clasificó correctamente a 62 personas como no enfermas, y efectivamente no tienen una enfermedad cardíaca.

- Verdaderos positivos (TP = 87): El modelo clasificó correctamente a 87 personas como enfermas, y en realidad sí tienen una enfermedad cardíaca.
- Falsos negativos (FN = 7): El modelo no detectó enfermedad en 7 personas, pero en realidad sí lo estaban. Esto es delicado en un problema médico, pues son pacientes con alguna enfermedad cardíaca no detectados por el modelo.
- Falsos positivos (FP = 26): El modelo clasificó incorrectamente a 26 personas como si tuvieran alguna enfermedad cardíaca, cuando en realidad no la tienen.

Extraemos las métricas

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Imprimir métricas
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8186813
```

```
cat("Precision (Precisión):", precision, "\n")
```

```
## Precision (Precisión): 0.7699115
```

```
cat("Recall (Sensibilidad):", recall, "\n")
```

```
## Recall (Sensibilidad): 0.9255319
```

```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.8405797
```

- Accuracy (Exactitud): El 81.86% de las predicciones fueron correctas.
- Precision (Precisión): El 76.99% de las veces que se predijo "Con Enfermedad", realmente era cierto.
- Recall (Sensibilidad): El modelo identifica correctamente el 92.55% de los pacientes que realmente tienen enfermedad.
- F1 Score: Es el promedio armónico entre precisión y recall, debido a que la precisión es un valor más bajo al ponderar quedó en 84.05% reflejando ciertos detalles para los falsos positivos.

BOSQUE ALEATORIO

Hiperparámetros

****Número de Árboles***

```

set.seed(123)
# Crear vector con diferentes números de árboles
n_pacientes <- seq(10, 300, by = 10)

# Vector vacío para almacenar F1 Score
f1_scores <- c()

# Entrenar bosques y calcular F1 para cada uno
for (n in n_pacientes) {
  cat("Entrenando con", n, "árboles...\n")

  bosque <- randomForest(Enfermedad_Cardíaca ~ ., data = enfermedad_train, ntree = n, importance = T, keep.inbag = T, oob.prox = T)

  # Obtener predicciones OOB
  predicciones_oob <- predict(bosque, type = "response")

  # Calcular F1
  f1 <- F1_Score(y_true = enfermedad_train$Enfermedad_Cardíaca, y_pred = predicciones_oob, positive = "Con Enfermedad")
  f1_scores <- c(f1_scores, f1)
}

```

```

## Entrenando con 10 árboles...
## Entrenando con 20 árboles...
## Entrenando con 30 árboles...
## Entrenando con 40 árboles...
## Entrenando con 50 árboles...
## Entrenando con 60 árboles...
## Entrenando con 70 árboles...
## Entrenando con 80 árboles...
## Entrenando con 90 árboles...
## Entrenando con 100 árboles...
## Entrenando con 110 árboles...
## Entrenando con 120 árboles...
## Entrenando con 130 árboles...
## Entrenando con 140 árboles...
## Entrenando con 150 árboles...
## Entrenando con 160 árboles...
## Entrenando con 170 árboles...
## Entrenando con 180 árboles...
## Entrenando con 190 árboles...
## Entrenando con 200 árboles...
## Entrenando con 210 árboles...
## Entrenando con 220 árboles...
## Entrenando con 230 árboles...
## Entrenando con 240 árboles...
## Entrenando con 250 árboles...
## Entrenando con 260 árboles...
## Entrenando con 270 árboles...
## Entrenando con 280 árboles...
## Entrenando con 290 árboles...
## Entrenando con 300 árboles...

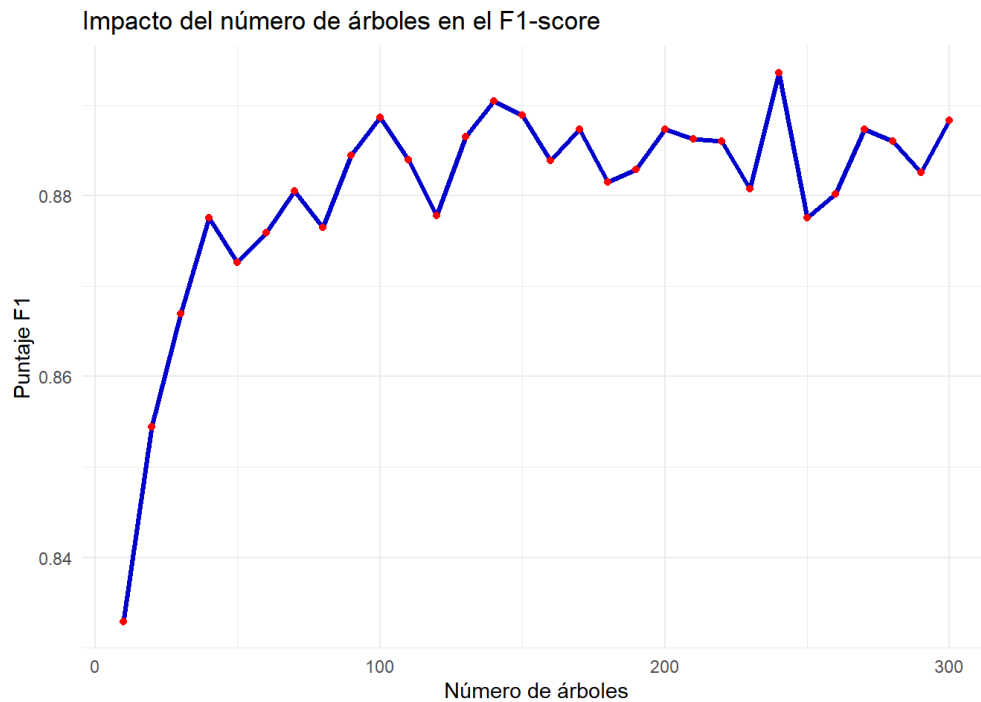
```

```

# Graficar F1 vs. número de árboles
df_f1 <- data.frame(n_pacientes, f1_scores)

ggplot(df_f1, aes(x = n_pacientes, y = f1_scores)) +
  geom_line(color = "blue3", linewidth = 1.2) +
  geom_point(color = "red", linewidth = 3) +
  labs(x = "Número de árboles", y = "Puntaje F1", title = "Impacto del número de árboles en el F1-score") +
  theme_minimal()

```



```
# Buscamos el F1 más alto
```

Podríamos decir que más árboles deberían ser capaces de producir un resultado más generalizado, pero al elegir un mayor número de árboles, la complejidad temporal del modelo también aumenta. En este gráfico, podemos ver claramente que el rendimiento del modelo aumenta drásticamente y luego se estanca en cierto nivel. Esto significa que elegir un gran número de estimadores en un modelo de bosque aleatorio no es la mejor idea. Si bien esto no degrada el modelo, puede ahorrarse complejidad computacional y una sobrecarga a la computadora. Notemos que con 240 árboles es suficiente para este modelo.

```
df_f1$n_pacientes[df_f1$f1_scores==max(df_f1$f1_scores)]
```

```
## [1] 240
```

Número de Nodos Terminales

```
set.seed(123)
# Definir los valores de max_depth
max_profundidad <- seq(2, 60, by = 2)
f1_scores <- c()

# Iterar sobre cada valor de max_depth
for (max_depth in max_profundidad) {
  cat("Entrenando con max_depth =", max_depth, "...\\n")

  # Entrenar el modelo con el valor de max_depth
  bosque <- randomForest(Enfermedad_Cardíaca ~ ., data = enfermedad_train, ntree = 240, maxnodes = max_depth, importance =
T, keep.inbag = T, oob.prox = T)

  # Obtener las predicciones fuera de bolsa
  predicciones_oob <- predict(bosque, type = "response")

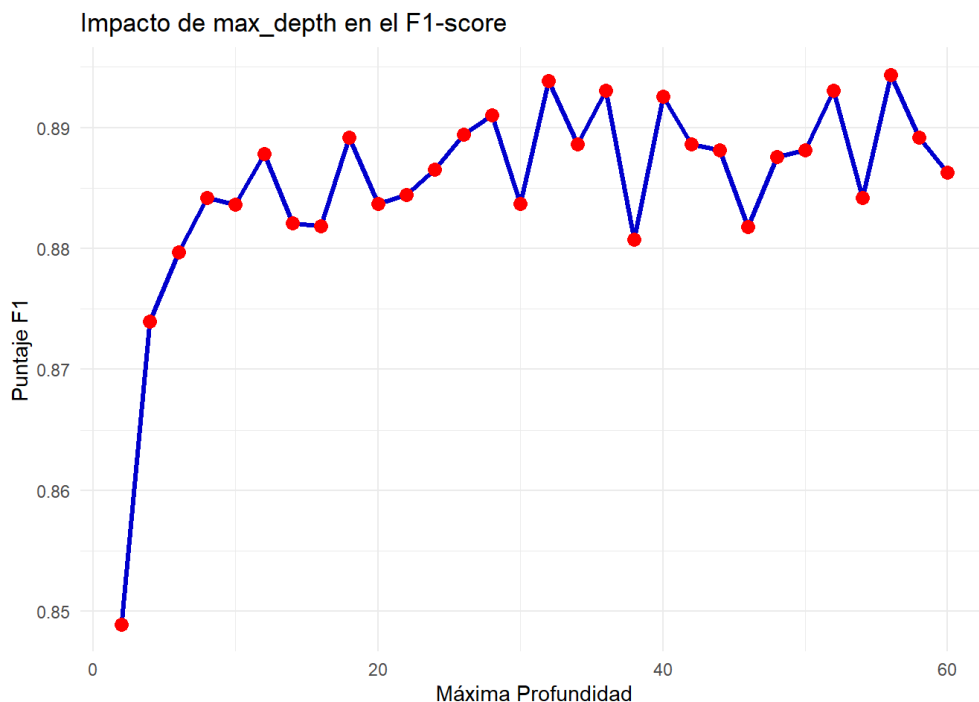
  # Calcular la matriz de confusión
  conf_matrix <- confusionMatrix(predicciones_oob, enfermedad_train$Enfermedad_Cardíaca, positive = "Con Enfermedad")

  # Extraer el F1-score de la matriz de confusión
  f1_scores <- c(f1_scores, conf_matrix$byClass["F1"])
}
```

```
## Entrenando con max_depth = 2 ...
## Entrenando con max_depth = 4 ...
## Entrenando con max_depth = 6 ...
## Entrenando con max_depth = 8 ...
## Entrenando con max_depth = 10 ...
## Entrenando con max_depth = 12 ...
## Entrenando con max_depth = 14 ...
## Entrenando con max_depth = 16 ...
## Entrenando con max_depth = 18 ...
## Entrenando con max_depth = 20 ...
## Entrenando con max_depth = 22 ...
## Entrenando con max_depth = 24 ...
## Entrenando con max_depth = 26 ...
## Entrenando con max_depth = 28 ...
## Entrenando con max_depth = 30 ...
## Entrenando con max_depth = 32 ...
## Entrenando con max_depth = 34 ...
## Entrenando con max_depth = 36 ...
## Entrenando con max_depth = 38 ...
## Entrenando con max_depth = 40 ...
## Entrenando con max_depth = 42 ...
## Entrenando con max_depth = 44 ...
## Entrenando con max_depth = 46 ...
## Entrenando con max_depth = 48 ...
## Entrenando con max_depth = 50 ...
## Entrenando con max_depth = 52 ...
## Entrenando con max_depth = 54 ...
## Entrenando con max_depth = 56 ...
## Entrenando con max_depth = 58 ...
## Entrenando con max_depth = 60 ...
```

```
# Graficar F1 vs. número de árboles
df_f2 <- data.frame(max_profundidad, f1_scores)

ggplot(df_f2, aes(x = max_profundidad, y = f1_scores)) +
  geom_line(color = "blue3", size = 1.2) +
  geom_point(color = "red", size = 3) +
  labs(x = "Máxima Profundidad", y = "Puntaje F1", title = "Impacto de max_depth en el F1-score") +
  theme_minimal()
```



La profundidad máxima de un árbol en Random Forest se define como la ruta más larga entre el nodo raíz y el nodo hoja; en este gráfico, podemos ver claramente que, a medida que aumenta la profundidad máxima del árbol de decisión, el rendimiento del modelo en el conjunto de entrenamiento aumenta continuamente. Por otro lado, a medida que aumenta el valor de max_depth, el rendimiento en el conjunto de prueba

aumenta inicialmente, pero a partir de cierto punto, comienza a disminuir periódicamente. La profundidad max es de 56.

```
df_f2$max_profundidad[df_f2$f1_scores==max(df_f2$f1_scores)]
```

```
## [1] 56
```

Máximo de Características

```
set.seed(123)
# Definir el rango de max_features
max_variables <- 1:(ncol(enfermedad_train) - 1) # -1 para excluir la columna de respuesta

# Inicializar un vector para almacenar los F1-scores
f1_scores <- c()

# Entrenar el modelo con diferentes valores de max_features
for (max_features in max_variables) {
  cat("Entrenando con max_features =", max_features, "...\\n")

  # Entrenar el randomForest con el valor de max_features
  bosque <- randomForest(enfermedad_train ~ ., data = enfermedad_train, ntree = 240, mtry = max_features, importance = T,
    keep.inbag = T, oob.prox = T)

  # Obtener las predicciones OOB
  predicciones_oob <- predict(bosque, type = "response")

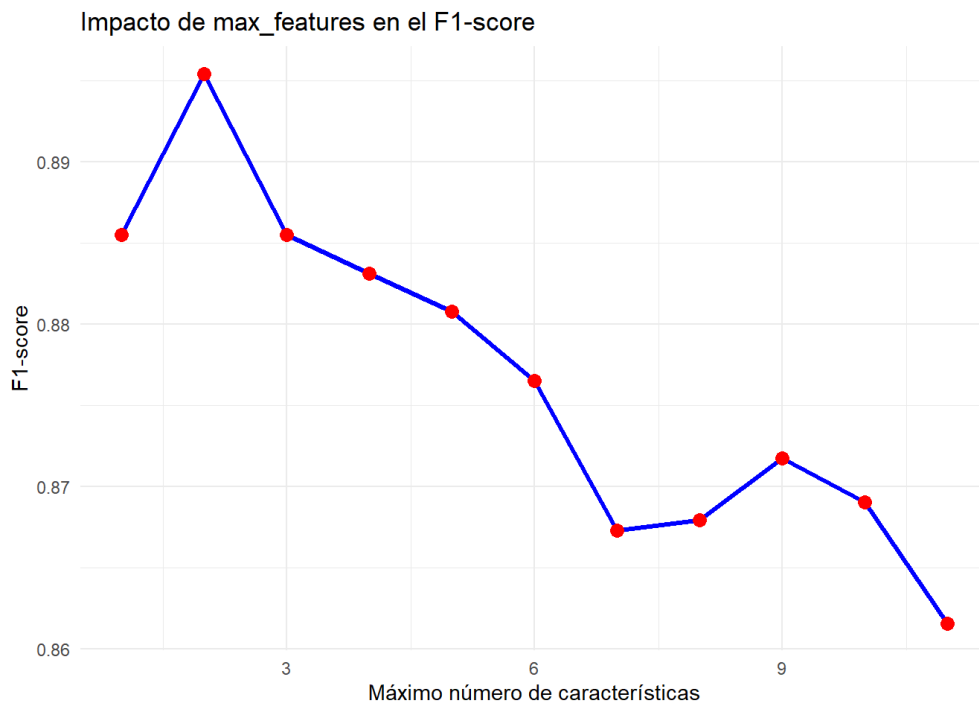
  # Calcular la matriz de confusión
  conf_matrix <- confusionMatrix(predicciones_oob, enfermedad_train$Enfermedad_Cardíaca, positive = "Con Enfermedad")

  # Calcular el F1-score
  f1_scores <- c(f1_scores, conf_matrix$byClass["F1"])
}
```

```
## Entrenando con max_features = 1 ...
## Entrenando con max_features = 2 ...
## Entrenando con max_features = 3 ...
## Entrenando con max_features = 4 ...
## Entrenando con max_features = 5 ...
## Entrenando con max_features = 6 ...
## Entrenando con max_features = 7 ...
## Entrenando con max_features = 8 ...
## Entrenando con max_features = 9 ...
## Entrenando con max_features = 10 ...
## Entrenando con max_features = 11 ...
```

```
# Crear un data frame para graficar
df_f3 <- data.frame(max_features = max_variables, f1_score = f1_scores)

# Graficar F1-score vs max_features
ggplot(df_f3, aes(x = max_features, y = f1_score)) +
  geom_line(color = "blue", lwd = 1.2) +
  geom_point(color = "red", size = 3) +
  labs(x = "Máximo número de características", y = "F1-score", title = "Impacto de max_features en el F1-score") +
  theme_minimal()
```



El número máximo de características proporcionadas a cada árbol en un bosque aleatorio. Sabemos que el bosque aleatorio selecciona muestras aleatorias de las características para encontrar la mejor distribución. Podemos observar que el rendimiento del modelo aumenta inicialmente a medida que aumenta el número de características máximas. Sin embargo, a partir de 3 características, la puntuación de prueba se satura e incluso comienza a disminuir hacia el final, lo que claramente indica que el modelo comienza a sobreajustarse. Con 2 características son más que suficientes, es recomendable considerar el valor predeterminado de este parámetro, que se establece como la raíz cuadrada del número de características presentes en el conjunto de datos.

```
df_f3$max_features[df_f3$f1_score==max(df_f3$f1_score)]
```

```
## [1] 2
```

Construcción del Modelo

Una vez tenemos las características necesarias para realizar el modelo de forma óptima, procedemos con él.

```
set.seed(123)
modelo1 <- randomForest(Enfermedad_Cardiaca ~ ., data = enfermedad_train, ntree=240, mtry=2, maxnodes=56) # Por defecto lo hace con gini
```

```
# Error OOB (ya viene incluido en el modelo)
modelo1$err.rate
```

##	OOB Sin Enfermedad	Con Enfermedad
##	[1,] 0.2158273	0.2195122 0.21290323
##	[2,] 0.2590090	0.2894737 0.23622047
##	[3,] 0.2509091	0.2946058 0.21682848
##	[4,] 0.2340426	0.2716981 0.20520231
##	[5,] 0.2162577	0.2261484 0.20867209
##	[6,] 0.2049780	0.2207358 0.19270833
##	[7,] 0.2022631	0.2265372 0.18341709
##	[8,] 0.1972222	0.2070064 0.18965517
##	[9,] 0.1774415	0.1930380 0.16545012
##	[10,] 0.1712329	0.1766562 0.16707022
##	[11,] 0.1839237	0.1812500 0.18599034
##	[12,] 0.1689373	0.1656250 0.17149758
##	[13,] 0.1646259	0.1619938 0.16666667
##	[14,] 0.1646259	0.1619938 0.16666667
##	[15,] 0.1551020	0.1526480 0.15700483
##	[16,] 0.1551020	0.1557632 0.15458937
##	[17,] 0.1523810	0.1464174 0.15700483
##	[18,] 0.1564626	0.1619938 0.15217391
##	[19,] 0.1591837	0.1588785 0.15942029
##	[20,] 0.1537415	0.1557632 0.15217391
##	[21,] 0.1510204	0.1651090 0.14009662
##	[22,] 0.1537415	0.1619938 0.14734300
##	[23,] 0.1537415	0.1713396 0.14009662
##	[24,] 0.1455782	0.1619938 0.13285024
##	[25,] 0.1413043	0.1583851 0.12801932
##	[26,] 0.1453804	0.1552795 0.13768116
##	[27,] 0.1399457	0.1583851 0.12560386
##	[28,] 0.1453804	0.1645963 0.13043478
##	[29,] 0.1399457	0.1583851 0.12560386
##	[30,] 0.1372283	0.1583851 0.12077295
##	[31,] 0.1290761	0.1552795 0.10869565
##	[32,] 0.1358696	0.1583851 0.11835749
##	[33,] 0.1358696	0.1583851 0.11835749
##	[34,] 0.1331522	0.1614907 0.11111111
##	[35,] 0.1304348	0.1614907 0.10628019
##	[36,] 0.1317935	0.1552795 0.11352657
##	[37,] 0.1372283	0.1645963 0.11594203
##	[38,] 0.1358696	0.1677019 0.11111111
##	[39,] 0.1345109	0.1677019 0.10869565
##	[40,] 0.1317935	0.1645963 0.10628019
##	[41,] 0.1345109	0.1677019 0.10869565
##	[42,] 0.1358696	0.1645963 0.11352657
##	[43,] 0.1358696	0.1708075 0.10869565
##	[44,] 0.1358696	0.1677019 0.11111111
##	[45,] 0.1290761	0.1552795 0.10869565
##	[46,] 0.1277174	0.1583851 0.10386473
##	[47,] 0.1277174	0.1583851 0.10386473
##	[48,] 0.1290761	0.1614907 0.10386473
##	[49,] 0.1304348	0.1614907 0.10628019
##	[50,] 0.1277174	0.1583851 0.10386473
##	[51,] 0.1263587	0.1583851 0.10144928
##	[52,] 0.1290761	0.1614907 0.10386473
##	[53,] 0.1290761	0.1614907 0.10386473
##	[54,] 0.1290761	0.1677019 0.09903382
##	[55,] 0.1290761	0.1645963 0.10144928
##	[56,] 0.1304348	0.1677019 0.10144928
##	[57,] 0.1277174	0.1645963 0.09903382
##	[58,] 0.1304348	0.1645963 0.10386473
##	[59,] 0.1290761	0.1614907 0.10386473
##	[60,] 0.1263587	0.1614907 0.09903382
##	[61,] 0.1290761	0.1614907 0.10386473
##	[62,] 0.1317935	0.1708075 0.10144928
##	[63,] 0.1317935	0.1677019 0.10386473
##	[64,] 0.1317935	0.1708075 0.10144928
##	[65,] 0.1358696	0.1739130 0.10628019
##	[66,] 0.1317935	0.1708075 0.10144928
##	[67,] 0.1331522	0.1739130 0.10144928
##	[68,] 0.1304348	0.1708075 0.09903382
##	[69,] 0.1290761	0.1708075 0.09661836

## [70,]	0.1304348	0.1739130	0.09661836
## [71,]	0.1277174	0.1708075	0.09420290
## [72,]	0.1277174	0.1708075	0.09420290
## [73,]	0.1277174	0.1739130	0.09178744
## [74,]	0.1250000	0.1677019	0.09178744
## [75,]	0.1277174	0.1708075	0.09420290
## [76,]	0.1290761	0.1739130	0.09420290
## [77,]	0.1250000	0.1677019	0.09178744
## [78,]	0.1263587	0.1708075	0.09178744
## [79,]	0.1263587	0.1708075	0.09178744
## [80,]	0.1263587	0.1708075	0.09178744
## [81,]	0.1277174	0.1739130	0.09178744
## [82,]	0.1277174	0.1739130	0.09178744
## [83,]	0.1290761	0.1739130	0.09420290
## [84,]	0.1290761	0.1770186	0.09178744
## [85,]	0.1290761	0.1739130	0.09420290
## [86,]	0.1317935	0.1801242	0.09420290
## [87,]	0.1331522	0.1832298	0.09420290
## [88,]	0.1304348	0.1801242	0.09178744
## [89,]	0.1304348	0.1801242	0.09178744
## [90,]	0.1304348	0.1770186	0.09420290
## [91,]	0.1331522	0.1801242	0.09661836
## [92,]	0.1317935	0.1770186	0.09661836
## [93,]	0.1317935	0.1770186	0.09661836
## [94,]	0.1304348	0.1739130	0.09661836
## [95,]	0.1304348	0.1770186	0.09420290
## [96,]	0.1331522	0.1801242	0.09661836
## [97,]	0.1358696	0.1832298	0.09903382
## [98,]	0.1317935	0.1832298	0.09178744
## [99,]	0.1304348	0.1770186	0.09420290
## [100,]	0.1317935	0.1770186	0.09661836
## [101,]	0.1331522	0.1801242	0.09661836
## [102,]	0.1317935	0.1770186	0.09661836
## [103,]	0.1304348	0.1739130	0.09661836
## [104,]	0.1277174	0.1708075	0.09420290
## [105,]	0.1277174	0.1708075	0.09420290
## [106,]	0.1277174	0.1739130	0.09178744
## [107,]	0.1277174	0.1770186	0.08937198
## [108,]	0.1277174	0.1770186	0.08937198
## [109,]	0.1263587	0.1739130	0.08937198
## [110,]	0.1290761	0.1770186	0.09178744
## [111,]	0.1290761	0.1770186	0.09178744
## [112,]	0.1290761	0.1770186	0.09178744
## [113,]	0.1290761	0.1770186	0.09178744
## [114,]	0.1277174	0.1739130	0.09178744
## [115,]	0.1277174	0.1739130	0.09178744
## [116,]	0.1277174	0.1739130	0.09178744
## [117,]	0.1263587	0.1677019	0.09420290
## [118,]	0.1277174	0.1770186	0.08937198
## [119,]	0.1277174	0.1770186	0.08937198
## [120,]	0.1250000	0.1708075	0.08937198
## [121,]	0.1250000	0.1708075	0.08937198
## [122,]	0.1250000	0.1708075	0.08937198
## [123,]	0.1250000	0.1708075	0.08937198
## [124,]	0.1222826	0.1677019	0.08695652
## [125,]	0.1236413	0.1708075	0.08695652
## [126,]	0.1236413	0.1708075	0.08695652
## [127,]	0.1236413	0.1708075	0.08695652
## [128,]	0.1236413	0.1708075	0.08695652
## [129,]	0.1222826	0.1677019	0.08695652
## [130,]	0.1236413	0.1708075	0.08695652
## [131,]	0.1222826	0.1645963	0.08937198
## [132,]	0.1236413	0.1677019	0.08937198
## [133,]	0.1236413	0.1645963	0.09178744
## [134,]	0.1236413	0.1645963	0.09178744
## [135,]	0.1222826	0.1645963	0.08937198
## [136,]	0.1222826	0.1645963	0.08937198
## [137,]	0.1236413	0.1645963	0.09178744
## [138,]	0.1222826	0.1645963	0.08937198
## [139,]	0.1222826	0.1645963	0.08937198
## [140,]	0.1222826	0.1645963	0.08937198

## [141,]	0.1222826	0.1645963	0.08937198
## [142,]	0.1222826	0.1645963	0.08937198
## [143,]	0.1250000	0.1645963	0.09420290
## [144,]	0.1263587	0.1677019	0.09420290
## [145,]	0.1250000	0.1677019	0.09178744
## [146,]	0.1250000	0.1677019	0.09178744
## [147,]	0.1236413	0.1645963	0.09178744
## [148,]	0.1236413	0.1645963	0.09178744
## [149,]	0.1236413	0.1645963	0.09178744
## [150,]	0.1236413	0.1645963	0.09178744
## [151,]	0.1250000	0.1677019	0.09178744
## [152,]	0.1250000	0.1677019	0.09178744
## [153,]	0.1236413	0.1645963	0.09178744
## [154,]	0.1222826	0.1614907	0.09178744
## [155,]	0.1195652	0.1614907	0.08695652
## [156,]	0.1222826	0.1645963	0.08937198
## [157,]	0.1209239	0.1614907	0.08937198
## [158,]	0.1222826	0.1645963	0.08937198
## [159,]	0.1222826	0.1677019	0.08695652
## [160,]	0.1250000	0.1677019	0.09178744
## [161,]	0.1236413	0.1677019	0.08937198
## [162,]	0.1222826	0.1677019	0.08695652
## [163,]	0.1236413	0.1708075	0.08695652
## [164,]	0.1236413	0.1708075	0.08695652
## [165,]	0.1236413	0.1708075	0.08695652
## [166,]	0.1250000	0.1708075	0.08937198
## [167,]	0.1236413	0.1708075	0.08695652
## [168,]	0.1250000	0.1739130	0.08695652
## [169,]	0.1250000	0.1739130	0.08695652
## [170,]	0.1277174	0.1739130	0.09178744
## [171,]	0.1250000	0.1708075	0.08937198
## [172,]	0.1250000	0.1708075	0.08937198
## [173,]	0.1250000	0.1708075	0.08937198
## [174,]	0.1250000	0.1708075	0.08937198
## [175,]	0.1263587	0.1739130	0.08937198
## [176,]	0.1277174	0.1708075	0.09420290
## [177,]	0.1263587	0.1708075	0.09178744
## [178,]	0.1277174	0.1739130	0.09178744
## [179,]	0.1277174	0.1739130	0.09178744
## [180,]	0.1263587	0.1708075	0.09178744
## [181,]	0.1250000	0.1708075	0.08937198
## [182,]	0.1290761	0.1739130	0.09420290
## [183,]	0.1277174	0.1708075	0.09420290
## [184,]	0.1290761	0.1708075	0.09661836
## [185,]	0.1331522	0.1770186	0.09903382
## [186,]	0.1290761	0.1677019	0.09903382
## [187,]	0.1277174	0.1677019	0.09661836
## [188,]	0.1263587	0.1677019	0.09420290
## [189,]	0.1304348	0.1708075	0.09903382
## [190,]	0.1290761	0.1708075	0.09661836
## [191,]	0.1317935	0.1708075	0.10144928
## [192,]	0.1317935	0.1708075	0.10144928
## [193,]	0.1304348	0.1708075	0.09903382
## [194,]	0.1345109	0.1770186	0.10144928
## [195,]	0.1331522	0.1739130	0.10144928
## [196,]	0.1317935	0.1739130	0.09903382
## [197,]	0.1317935	0.1708075	0.10144928
## [198,]	0.1304348	0.1739130	0.09661836
## [199,]	0.1331522	0.1739130	0.10144928
## [200,]	0.1317935	0.1770186	0.09661836
## [201,]	0.1317935	0.1770186	0.09661836
## [202,]	0.1317935	0.1770186	0.09661836
## [203,]	0.1317935	0.1770186	0.09661836
## [204,]	0.1317935	0.1739130	0.09903382
## [205,]	0.1304348	0.1739130	0.09661836
## [206,]	0.1317935	0.1770186	0.09661836
## [207,]	0.1317935	0.1770186	0.09661836
## [208,]	0.1331522	0.1770186	0.09903382
## [209,]	0.1317935	0.1739130	0.09903382
## [210,]	0.1331522	0.1770186	0.09903382
## [211,]	0.1331522	0.1770186	0.09903382

```
## [212,] 0.1317935      0.1770186      0.09661836
## [213,] 0.1317935      0.1801242      0.09420290
## [214,] 0.1331522      0.1770186      0.09903382
## [215,] 0.1304348      0.1770186      0.09420290
## [216,] 0.1304348      0.1770186      0.09420290
## [217,] 0.1290761      0.1739130      0.09420290
## [218,] 0.1304348      0.1770186      0.09420290
## [219,] 0.1277174      0.1708075      0.09420290
## [220,] 0.1290761      0.1739130      0.09420290
## [221,] 0.1263587      0.1708075      0.09178744
## [222,] 0.1290761      0.1770186      0.09178744
## [223,] 0.1304348      0.1801242      0.09178744
## [224,] 0.1290761      0.1770186      0.09178744
## [225,] 0.1290761      0.1770186      0.09178744
## [226,] 0.1277174      0.1739130      0.09178744
## [227,] 0.1290761      0.1770186      0.09178744
## [228,] 0.1290761      0.1770186      0.09178744
## [229,] 0.1290761      0.1770186      0.09178744
## [230,] 0.1290761      0.1770186      0.09178744
## [231,] 0.1290761      0.1770186      0.09178744
## [232,] 0.1277174      0.1739130      0.09178744
## [233,] 0.1277174      0.1739130      0.09178744
## [234,] 0.1277174      0.1739130      0.09178744
## [235,] 0.1277174      0.1739130      0.09178744
## [236,] 0.1290761      0.1770186      0.09178744
## [237,] 0.1290761      0.1770186      0.09178744
## [238,] 0.1277174      0.1770186      0.08937198
## [239,] 0.1277174      0.1770186      0.08937198
## [240,] 0.1277174      0.1770186      0.08937198
```

```
exactitud_oob <- 1 - modelo1$err.rate[which.min(modelo1$err.rate[, "Oob"])]
cat("La exactitud OOB es:", round(exactitud_oob * 100, 1), "%\n")
```

```
## La exactitud OOB es: 88 %
```

- Conforme se agregan más árboles, el bosque mejora su capacidad de generalización, aprox. después del árbol 148, el error OOB (porcentaje de predicciones correctas sobre los datos OOB) se estabiliza y deja de mejorar, usualmente oscilando el 0.125000.
- Una OOB accuracy del 88% quiere decir que el modelo predice correctamente el diagnóstico (enfermedad cardíaca o no) en el 88% de los casos del conjunto de datos fuera de entrenamiento.

```
head(modelo1$votes)
```

```
##      Sin Enfermedad Con Enfermedad
## 3      1.00000000      0.00000000
## 5      1.00000000      0.00000000
## 6      0.00000000      1.00000000
## 8      0.04210526      0.9578947
## 10     1.00000000      0.00000000
## 11     0.04444444      0.9555556
```

```
tail(modelo1$votes)
```

```
##      Sin Enfermedad Con Enfermedad
## 908     0.90109890      0.09890110
## 909     0.97916667      0.02083333
## 910     0.77011494      0.22988506
## 913     0.95238095      0.04761905
## 915     0.05319149      0.94680851
## 916     0.06097561      0.93902439
```

- Notemos que las clasificaciones son consistentes y no hay sesgo respecto al como se clasifica el estado del paciente.

Realizamos las predicciones

```
# Predicciones OOB
predicciones_oob <- modelo1$predicted
```

Matriz de Confusión

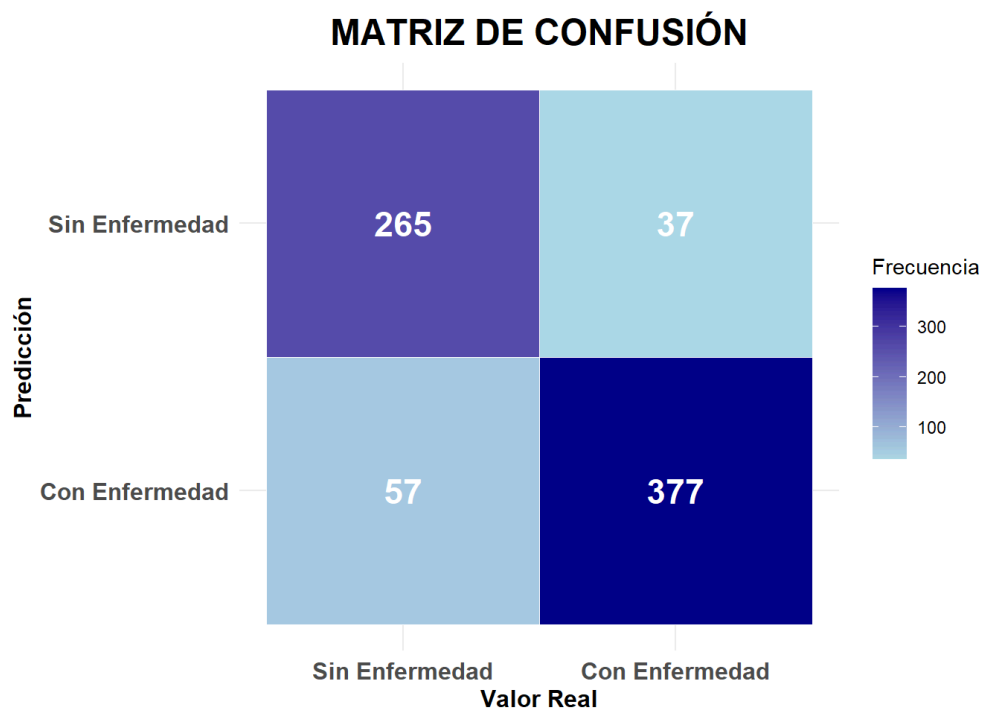
```
reales <- enfermedad_train$Enfermedad_Cardiaca # Valores reales
# Matriz de confusión
conf_matrix <- confusionMatrix(data = predicciones_oob, reference = reales, positive = "Con Enfermedad")
print(conf_matrix$table)
```

```
##               Reference
## Prediction      Sin Enfermedad Con Enfermedad
## Sin Enfermedad      265           37
## Con Enfermedad      57           377
```

```
cm_table <- as.data.frame(conf_matrix$table)

# Invertir el orden de los niveles del eje Y
cm_table$Prediction <- factor(cm_table$Prediction, levels = rev(levels(cm_table$Prediction)))

ggplot(cm_table, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), color = "white", size = 6, fontface = "bold") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name="Frecuencia") +
  labs(
    title = "MATRIZ DE CONFUSIÓN",
    x = "Valor Real",
    y = "Predicción"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    axis.text = element_text(size = 12, face="bold"),
    axis.title = element_text(size = 12, face="bold")
  )
```



- Verdaderos negativos (TN = 265): El modelo clasificó correctamente a 266 personas como no enfermas, y efectivamente no tienen una enfermedad cardiaca.
 - Verdaderos positivos (TP = 377): El modelo clasificó correctamente a 378 personas como enfermas, y en realidad sí tienen una enfermedad cardiaca.
 - Falsos negativos (FN = 37): El modelo no detectó enfermedad en 36 personas, pero en realidad sí lo estaban. Esto es delicado en un problema médico, pues son pacientes con alguna enfermedad cardiaca no detectados por el modelo.

- Falsos positivos (FP = 57): El modelo clasificó incorrectamente a 56 personas como si tuvieran alguna enfermedad cardíaca, cuando en realidad no la tienen.

Métricas

```
# Extraer métricas
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1 <- conf_matrix$byClass["F1"]

# Mostrar resultados
cat("Accuracy (Exactitud):", accuracy, "\n")
```

```
## Accuracy (Exactitud): 0.8722826
```

```
cat("Precision (Precisión): ", round(precision, 4), "\n")
```

```
## Precision (Precisión): 0.8687
```

```
cat("Recall (Sensibilidad): ", round(recall, 4), "\n")
```

```
## Recall (Sensibilidad): 0.9106
```

```
cat("Puntaje F1: ", round(f1, 4), "\n")
```

```
## Puntaje F1: 0.8892
```

- Accuracy (Exactitud) = 87.22%: El 86.95% de todas las predicciones del modelo fueron correctas, tanto para personas con una enfermedad cardíaca como sin ella.
 - Precision (Precisión) = 86.87%: De todas las personas que el modelo clasificó como enfermas, el 86.87% realmente lo eran. Esta métrica es importante para evitar falsos positivos, es decir, no generar diagnósticos erróneos en personas sanas.
 - Recall (Sensibilidad) = 91.06%: El modelo fue capaz de detectar correctamente el 91.06% de los verdaderos casos de enfermedad cardíaca. Esta métrica es clave en el área médica, porque queremos identificar a la mayor cantidad de pacientes enfermos posible. Aprox un 10% de los casos no fueron detectados (falsos negativos).
 - F1 Score = 88.92%: Este es un balance entre precisión y recall. Un F1 alto indica que el modelo tiene un buen compromiso entre detectar correctamente a quienes están enfermos y no clasificar erróneamente a quienes no lo están.

CONCLUSIONES

Comparación de Modelos

En la siguiente tabla se puede observar los valores de la matriz de confusión, métricas más importantes y las ventajas o limitaciones que presenta cada uno de los 4 modelos finales que se utilizaron para clasificar a los pacientes que sí presentan la enfermedad cardíaca, esto nos ayudará a elegir el modelo que mejor se ajusta a los objetivos iniciales.

Métrica	Logístico	QDA	Árbol de Decisión	Bosque Aleatorio
Verdaderos Negativos (TN)	86	84	87	377
Falsos Positivos (FP)	17	16	26	57
Falsos Negativos (FN)	8	10	7	37
Verdaderos Positivos (TP)	71	72	62	265
Exactitud (Accuracy)	86.26%	85.71%	81.86%	87.22%
Precisión (Precision)	83.49%	84%	76.99%	86.87%
Sensibilidad (Recall)	91.48%	89.36%	92.55%	91.06%
F1 Score	87.30%	86.59%	84.05%	88.92%
Ventajas	Interpretación de Variables	Equilibrado	Mejor Recall	Superior en Métricas
Limitaciones	Suponer relación lineal	Supuestos	Errores (falsos positivos)	Visualización y Costo

Selección del Mejor Modelo

Tras evaluar múltiples enfoques predictivos sobre el diagnóstico de enfermedad cardíaca, se concluye lo siguiente: El Bosque Aleatorio demostró ser el modelo con mejor rendimiento global, destacando en precisión, sensibilidad y F1 Score. Su capacidad para manejar relaciones no lineales, interacciones complejas y su resistencia al sobreajuste lo hacen ideal para aplicaciones clínicas automatizadas. La Regresión Logística es una excelente alternativa cuando se prioriza la interpretabilidad y el análisis de factores clínicos individuales. Aunque su desempeño fue ligeramente inferior al Random Forest, su facilidad de implementación y explicación la hacen muy valiosa en la práctica médica. Los modelos basados en Árboles de Decisión, en especial el con validación cruzada, mostraron un buen equilibrio entre interpretabilidad y rendimiento. Son ideales cuando se busca visualizar el proceso de decisión, aunque con menor precisión que los modelos de ensamble. El QDA funcionó bien, pero su uso está condicionado al cumplimiento de supuestos estadísticos (normalidad y heterocedasticidad), lo que puede limitar su aplicabilidad en datos reales clínicos. En síntesis, aunque todos los modelos ofrecieron buenos resultados, el Bosque Aleatorio representa la mejor combinación de sensibilidad, precisión y robustez, especialmente importante en contextos donde los errores de diagnóstico pueden tener consecuencias graves.

Conclusiones y Recomendaciones

Este trabajo representó una oportunidad valiosa para integrar conocimientos estadísticos, clínicos y computacionales en el análisis predictivo de una problemática médica relevante como es la enfermedad cardíaca. A través de las distintas etapas del proyecto desde la limpieza y exploración de los datos hasta la evaluación de modelos predictivos fue posible comprender cómo las técnicas estadísticas y de aprendizaje automático pueden aportar significativamente en contextos reales, donde las decisiones tienen impacto directo sobre la salud de las personas. El proceso permitió observar que, más allá de la aplicación técnica de los modelos, es fundamental contar con una base de datos bien estructurada y clínicamente coherente. El análisis cuidadoso de las variables, el tratamiento adecuado de las transformaciones y la validación del comportamiento de los algoritmos fueron aspectos centrales para asegurar la confiabilidad de los resultados. También se hizo evidente que la elección de un modelo no puede basarse únicamente en métricas de desempeño: la interpretabilidad, la robustez frente al ruido, la facilidad de implementación y el contexto de uso deben ser considerados al momento de seleccionar una herramienta predictiva. En conjunto, este estudio refleja cómo la estadística, aplicada de manera crítica y contextualizada, puede apoyar de forma efectiva el diagnóstico clínico. La tecnología no reemplaza al criterio médico, pero sí lo complementa, ofreciendo una visión basada en evidencia y datos objetivos. A medida que estas herramientas se integren más en el ámbito de la salud, será clave mantener una mirada ética, crítica y orientada al beneficio del paciente.