

¿Cómo realizar la limpieza y análisis de datos?

Autores: Eduardo Mora González y Diego Sánchez De La Fuente

Enero 2023

Contents

CARGA DEL FICHERO DE DATOS	2
Descripción del dataset:	2
Preprocesado y gestión de características	4
Valores nulos del conjunto de los datos	4
Normalización del conjunto de los datos	5
Construcción de conjunto de datos final	18
Correlaciones	22
Análisis de componentes principales (PCA)	24
Exportación de los datos	33
Análisis de los datos	33
COMprobación de la normalidad y la homogeneidad de la varianza	33
Pruebas estadísticas	39
¿Que variables cuantitativas ejercen mayor influencia en la variable que define si hay una enfermedad cardíaca	39
Grupos de datos	40
¿La Presión arterial y su influencia en la enfermedad cardíaca?	41
Modelo Arbol de Decision	41
Test estadísticos de significancia	41
Aplicación del modelo Decision Tree (Arbol de decisión)	47
Evaluación del modelo arbol de decision	51

Instalamos y cargamos las librerías necesarias.

```
if (!require('readr')) install.packages('readr'); library('readr')
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('DataExplorer')) install.packages('DataExplorer'); library('DataExplorer')
if (!require('corrplot')) install.packages("corrplot"); library(corrplot)
if (!require('factoextra')) install.packages("factoextra"); library(factoextra)
if (!require('dplyr')) install.packages("dplyr"); library(dplyr)
```

CARGA DEL FICHERO DE DATOS

```
datos <- read_csv("./fichero_original_datos.csv")
attach(datos)
```

Descripción del dataset:

El conjunto de datos ha sido extraído de Kaggle: <https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset>, está compuesto de 12 variables y 918 registros. Que correlacionan una serie de características recogidas de varios pacientes con la posibilidad de sufrir un ataque al corazón.

Explicación de cada variable:

- **Age:** Edad del paciente
- **Sex:** Sexo del paciente
- **ChestPainType:** Tipo de dolor torácico: Angina Típica Angina Atípica Dolor no debido a una angina Asintomático
- **RestingBP:** Presión arterial en reposo (en mm Hg)
- **Cholesterol:** Colesterol en sangre (mg/dL)
- **FastingBS:** Tiene Glucemia en ayunas > 120 mg/dl -> (1: True, 0: False)
- **RestingECG:** Resultados electrocardiográficos en reposo Value 0: normal Value 1: Tiene anomalía de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST > 0.05 mV) Value 2 Muestra hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes
- **MaxHR:** Frecuencia cardíaca máxima alcanzada
- **ExerciseAngina:** Angina inducida por el ejercicio (1 = sí, 0 = no)
- **Oldpeak:** Descenso del segmento ST inducido por el ejercicio en relación con el reposo ('Segmento ST' se relaciona con las posiciones en el gráfico de Electro cardiograma).
- **ST_Slope:** La pendiente del segmento ST de ejercicio máximo: 0: pendiente descendente 1: plano 2: pendiente ascendente
- **HeartDisease:** Variable Objetivo: 0= menos posibilidades de infarto 1= más posibilidades de infarto.

Tipo de dato asignado a cada campo:

```
# Cargamos en un vector los tipos de variable del dataset
vector_tipos <- sapply(datos, function(x) class(x))
print(vector_tipos)
```

```
##           Age           Sex ChestPainType           RestingBP           Cholesterol
```

```
##      "numeric"      "character"      "character"      "numeric"      "numeric"
##      FastingBS      RestingECG          MaxHR ExerciseAngina      Oldpeak
##      "numeric"      "character"      "numeric"      "character"      "numeric"
##      ST_Slope      HeartDisease
##      "character"      "numeric"
```

Ahora vamos a ver la estructura del juego de datos

```
str(datos)
```

```
## spec_tbl_df [918 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ Age          : num [1:918] 40 49 37 48 54 39 45 54 37 48 ...
## $ Sex          : chr [1:918] "M" "F" "M" "F" ...
## $ ChestPainType : chr [1:918] "ATA" "NAP" "ATA" "ASY" ...
## $ RestingBP    : num [1:918] 140 160 130 138 150 120 130 110 140 120 ...
## $ Cholesterol  : num [1:918] 289 180 283 214 195 339 237 208 207 284 ...
## $ FastingBS    : num [1:918] 0 0 0 0 0 0 0 0 0 0 ...
## $ RestingECG   : chr [1:918] "Normal" "Normal" "ST" "Normal" ...
## $ MaxHR        : num [1:918] 172 156 98 108 122 170 170 142 130 120 ...
## $ ExerciseAngina: chr [1:918] "N" "N" "N" "Y" ...
## $ Oldpeak      : num [1:918] 0 1 0 1.5 0 0 0 0 1.5 0 ...
## $ ST_Slope     : chr [1:918] "Up" "Flat" "Up" "Flat" ...
## $ HeartDisease : num [1:918] 0 1 0 1 0 0 0 0 1 0 ...
## - attr(*, "spec")=
## .. cols(
## ..   Age = col_double(),
## ..   Sex = col_character(),
## ..   ChestPainType = col_character(),
## ..   RestingBP = col_double(),
## ..   Cholesterol = col_double(),
## ..   FastingBS = col_double(),
## ..   RestingECG = col_character(),
## ..   MaxHR = col_double(),
## ..   ExerciseAngina = col_character(),
## ..   Oldpeak = col_double(),
## ..   ST_Slope = col_character(),
## ..   HeartDisease = col_double()
## .. )
## - attr(*, "problems")=<externalptr>
```

Vamos ahora a sacar estadísticas básicas

```
summary(datos)
```

```
##      Age          Sex          ChestPainType      RestingBP
## Min.   :28.00    Length:918    Length:918      Min.    : 0.0
## 1st Qu.:47.00    Class :character  Class :character 1st Qu.:120.0
## Median :54.00    Mode  :character  Mode  :character Median :130.0
## Mean   :53.51
## 3rd Qu.:60.00
## Max.   :77.00
## Cholesterol      FastingBS      RestingECG      MaxHR
##
```

```
## Min. : 0.0 Min. :0.0000 Length:918 Min. : 60.0
## 1st Qu.:173.2 1st Qu.:0.0000 Class :character 1st Qu.:120.0
## Median :223.0 Median :0.0000 Mode :character Median :138.0
## Mean :198.8 Mean :0.2331 Mean :136.8
## 3rd Qu.:267.0 3rd Qu.:0.0000 3rd Qu.:156.0
## Max. :603.0 Max. :1.0000 Max. :202.0
## ExerciseAngina Oldpeak ST_Slope HeartDisease
## Length:918 Min. :-2.6000 Length:918 Min. :0.0000
## Class :character 1st Qu.: 0.0000 Class :character 1st Qu.:0.0000
## Mode :character Median : 0.6000 Mode :character Median :1.0000
## Mean : 0.8874 Mean :0.5534
## 3rd Qu.: 1.5000 3rd Qu.:1.0000
## Max. : 6.2000 Max. :1.0000
```

Observamos los primeros 5 registros:

```
head(datos, 5L)
```

```
## # A tibble: 5 x 12
##   Age Sex ChestPainT~1 Resti~2 Chole~3 Fasti~4 Resti~5 MaxHR Exerc~6 Oldpeak
##   <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <chr> <dbl>
## 1 40 M ATA 140 289 0 Normal 172 N 0
## 2 49 F NAP 160 180 0 Normal 156 N 1
## 3 37 M ATA 130 283 0 ST 98 N 0
## 4 48 F ASY 138 214 0 Normal 108 Y 1.5
## 5 54 M NAP 150 195 0 Normal 122 N 0
## # ... with 2 more variables: ST_Slope <chr>, HeartDisease <dbl>, and
## # abbreviated variable names 1: ChestPainType, 2: RestingBP, 3: Cholesterol,
## # 4: FastingBS, 5: RestingECG, 6: ExerciseAngina
```

Preprocesado y gestión de características

Valores nulos del conjunto de los datos

De tipo numérico

```
colSums(is.na(datos))
```

```
##           Age           Sex ChestPainType RestingBP Cholesterol
##           0           0           0           0           0
## FastingBS RestingECG MaxHR ExerciseAngina Oldpeak
##           0           0           0           0           0
## ST_Slope HeartDisease
##           0           0
```

De tipo cadena

```
colSums(datos=="")
```

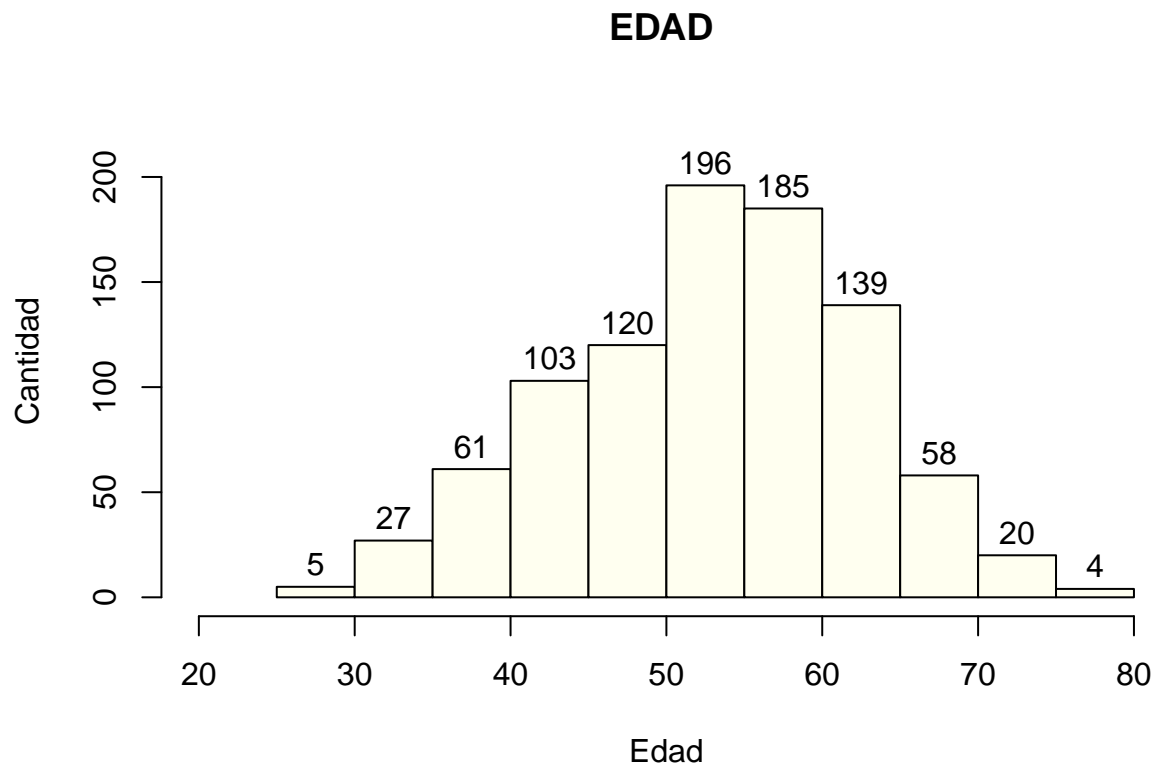
```
##           Age           Sex ChestPainType      RestingBP      Cholesterol
##           0             0           0           0           0
##      FastingBS      RestingECG      MaxHR ExerciseAngina      Oldpeak
##           0             0           0           0           0
##      ST_Slope      HeartDisease
##           0             0
```

Como se puede comprobar, tenemos la “suerte” de no tener ningún valor nulo o vacío en los dos juegos de datos.

Normalización del conjunto de los datos

EDAD

```
#Histograma de la característica edad del primer conjunto de datos
h1 <- hist(datos$Age, xlab="Edad", col="ivory",
           ylab="Cantidad", main="EDAD ", ylim = c(0, 225), xlim = c(20,80))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```



Como se puede observar, la franja de entre los 50 y 60 años son donde más datos existen, mientras que los extremos donde menos datos.

SEXO

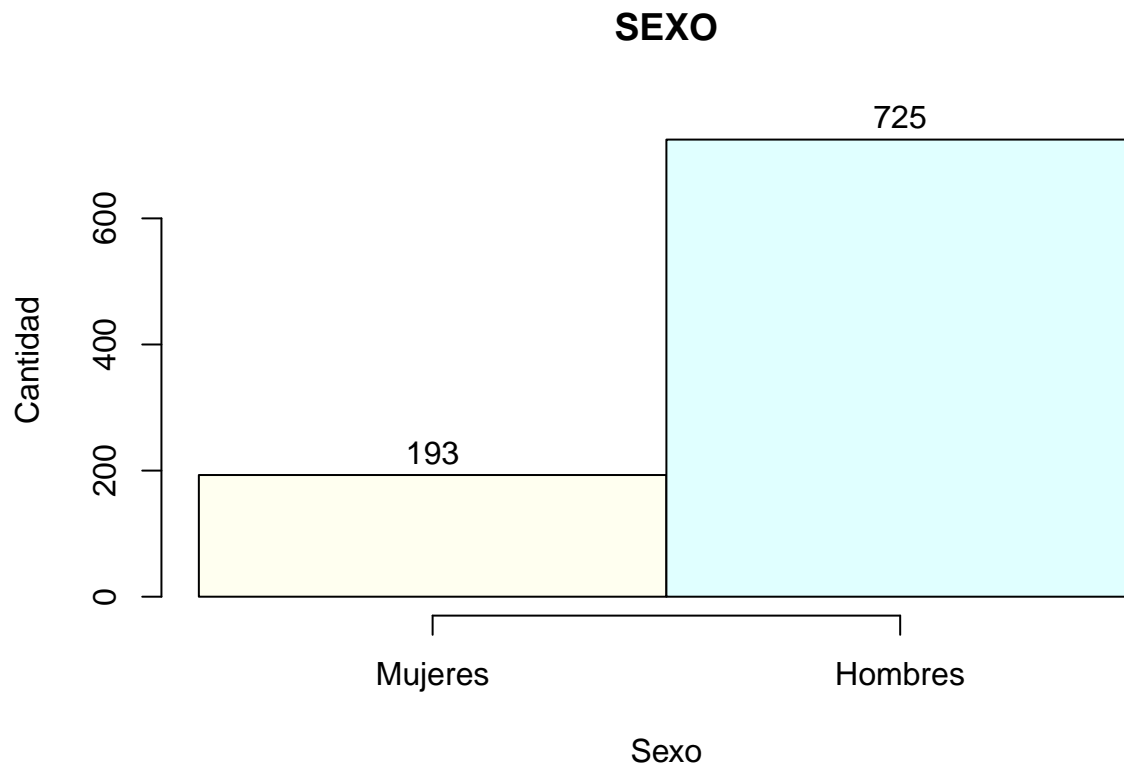
Normalizamos para tenerlo de tipo numérico todas la variables

```
#Cambiamos las letras por los números
datos$Sex [datos$Sex == "M"] <- 1
datos$Sex [datos$Sex == "F"] <- 0

#Pasamos de carácter a numérico
datos$Sex <- as.numeric(datos$Sex)
```

Una vez normalizada la característica , analizamos el conjunto de los datos contemplados en esta.

```
h1 <- hist(datos$Sex, xlab="Sexo", col=c("ivory", "lightcyan"),
          ylab="Cantidad", main="SEXO", breaks = 2, ylim = c(0, 750), axes = FALSE)
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75), cex.axis=1, labels = c("Mujeres","Hombres" ))
axis(2)
```



TIPO DE DOLOR TORÁCICO (ChestPainType)

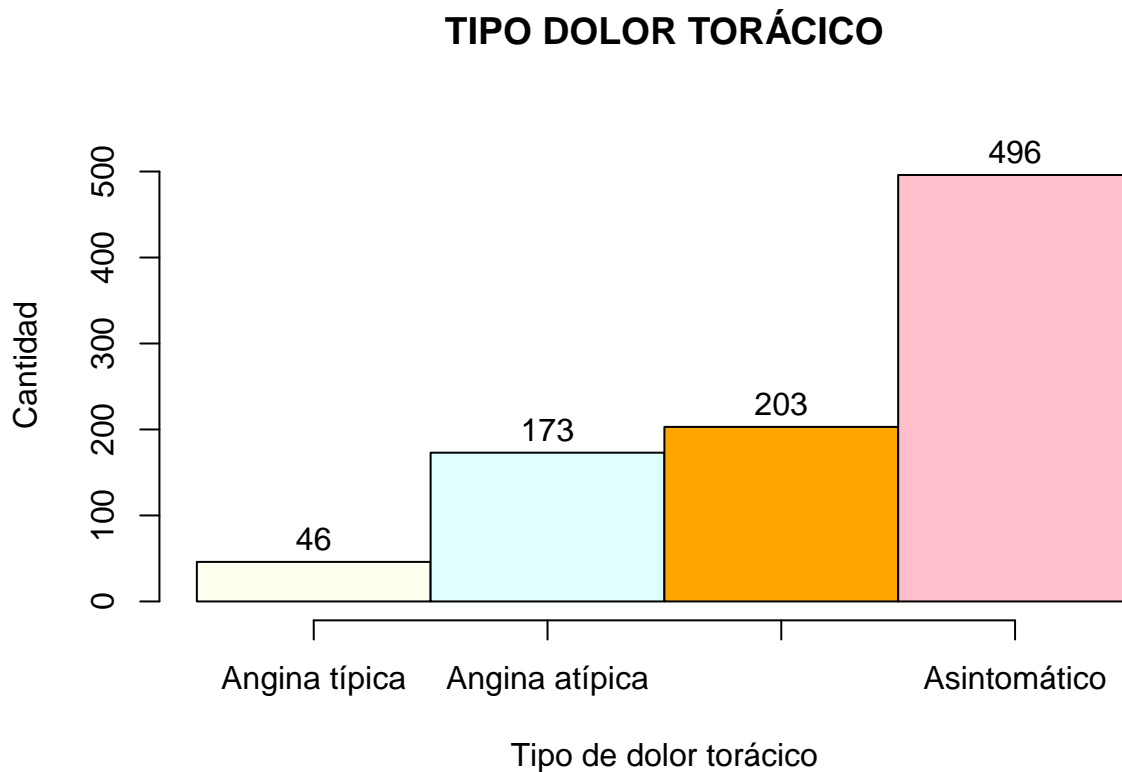
Nos damos cuenta de que el conjunto de datos viene identificado por 4 variables categóricas (TA: angina típica, ATA: angina atípica, NAP: dolor no anginal, ASY: asintomático). Normalizamos para tenerlo de tipo numérico todas la variables:

```
#Cambiamos las letras por los números
datos$ChestPainType [datos$ChestPainType == "TA"] <- 0
datos$ChestPainType [datos$ChestPainType == "ATA"] <- 1
datos$ChestPainType [datos$ChestPainType == "NAP"] <- 2
datos$ChestPainType [datos$ChestPainType == "ASY"] <- 3

#Pasamos de carácter a numérico
datos$ChestPainType <- as.numeric(datos$ChestPainType)
```

Una vez normalizada la característica , analizamos el conjunto de los datos contemplados en esta.

```
h1 <- hist(datos$ChestPainType, xlab="Tipo de dolor torácico",
           col= c("ivory", "lightcyan", "ORANGE", "PINK"),
           ylab="Cantidad", main="TIPO DOLOR TORÁCICO",
           ylim = c(0, 550), axes = FALSE,
           breaks=seq(min(datos$ChestPainType)-0.5,
                      max(datos$ChestPainType)+0.5, by=1) )
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0,1,2,3), cex.axis=1,
     labels = c("Angina típica", "Angina atípica","Dolor no anginal", "Asintomático" ))
axis(2)
```



como se puede comprobar, tenemos mas casos de de asintomaticos que del resto.

PRESIÓN ARTERIAL EN REPOSO (RestingBP)

Como se muestran en las estadísticas esta característica es de tipo numérico y en el conjunto de datos va desde 0 hasta 200. Como se puede apreciar, tener una presión arterial de 0 es estar considerado muerto, por lo que considero que el valor 0 es un valor nulo.

Lo primero que se va a hacer es obtener el número de casos que la presión arterial es 0, y se consideraran las diversas formas de tratar estos datos:

```
#Veces que aparece el valor cero en la presion arterial
length(datos$RestingBP[datos$RestingBP == 0])
```

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

Como solo aparece una vez, se le asignará un valor por defecto. El valor por defecto será el más común.

```
#Función para calcular el valor más común
common_value <- function(x) {
  uniqx <- unique(na.omit(x))
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

#Calculamos el valor más comun
BP_comun <- common_value(datos$RestingBP)

#Asignamos el valor
datos$RestingBP[datos$RestingBP == 0] <- BP_comun

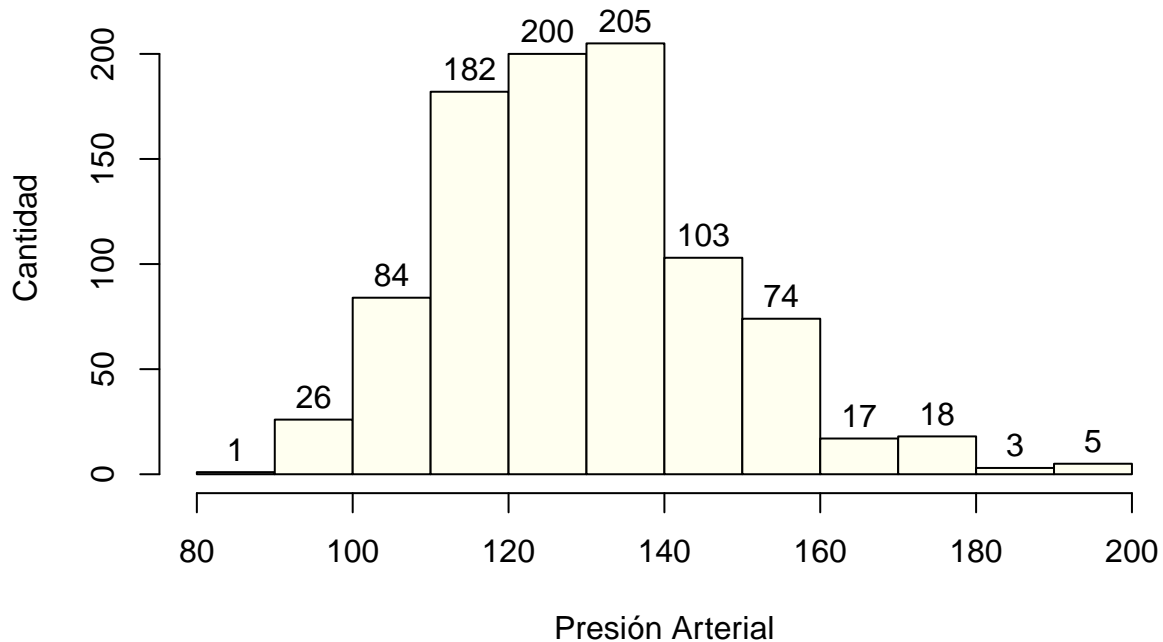
#vemos las estadisticas del dato
summary(datos$RestingBP)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      80.0   120.0   130.0   132.5   140.0   200.0
```

Ahora ya tenemos los valores entre 80 y 200 que son un rango normal para estos valores.

```
#Histograma de la característica Presión Arterial del primer conjunto de datos
h1 <- hist(datos$RestingBP, xlab="Presión Arterial", col="ivory",
           ylab="Cantidad", main="PRESIÓN ARTERIAL EN REPOSO",
           ylim = c(0, 225), xlim = c(80,200))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```


PRESIÓN ARTERIAL EN REPOSO



COLESTEROL (Cholesterol)

La siguiente característica es de tipo numérico. Al igual que en la presión arterial en reposo, que tenemos valores 0 que debemos analizar. Lo primero que se va a hacer es obtener el numero de casos que el colesterol es 0, y se consideraran las diversas formas de tratar estos datos.

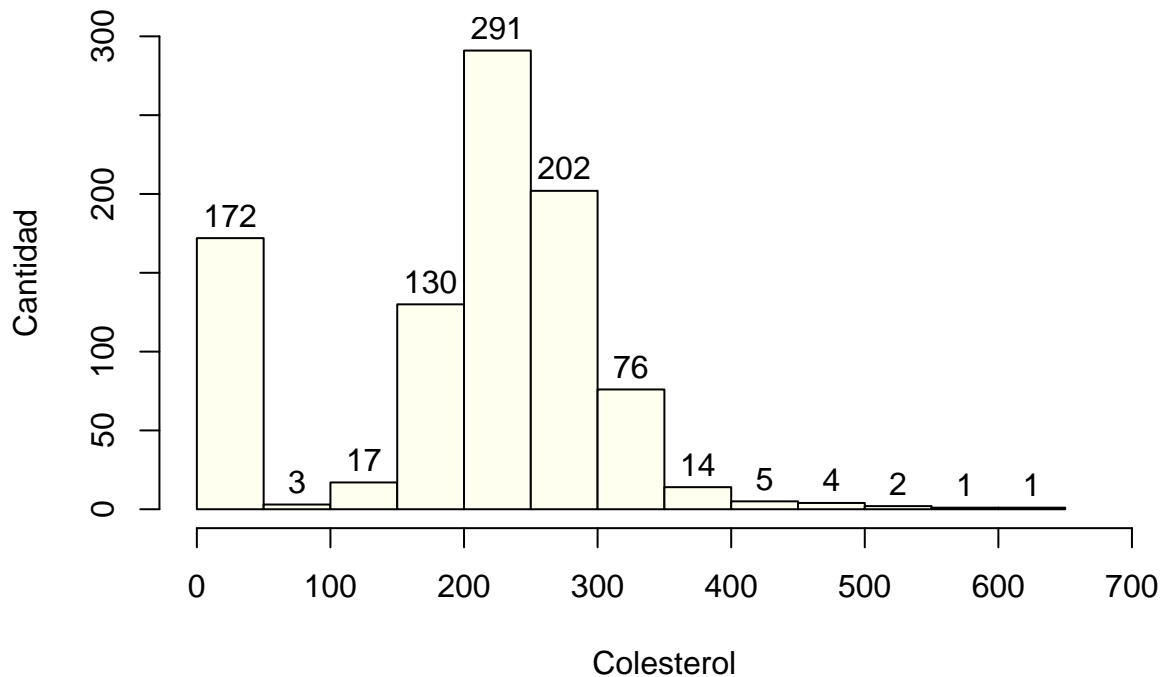
```
#Veces que aparece el valor cero en la presion arterial
length(datos$RestingBP[datos$Cholesterol == 0])
```

```
## [1] 172
```

Esta vez tenemos 172 casos en lo que ocurre esto (equivale a un 18% de los casos totales). Antes de ver que valor se le asignan, se va a graficar los datos para ver de manera grafica que opción tomar: el valor medio o el más común.

```
h1 <- hist(datos$Cholesterol, xlab="Cholesterol", col="ivory",
           ylab="Cantidad", main="COLESTEROL SIN TRATAR NULOS", ylim = c(0,300),
           xlim = c(0, 700))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```

COLESTEROL SIN TRATAR NULOS



Tras analizar la gráfica y para no perder estos datos, se le asignaran un valor por defecto, que será la media de los datos. Esta decisión se ha tomado ya que poner el más común, nos crearía un conjunto de datos muy distintos entre unas medidas y otras, mientras que poner la media sería un valor que tenga en cuenta el grueso de todos los datos.

```
#Calculamos el valor más comun
colesterol_media <- mean(datos$Cholesterol)

#Asignamos el valor truncado para evitar decimales
datos$Cholesterol[datos$Cholesterol == 0] <- trunc(colesterol_media)

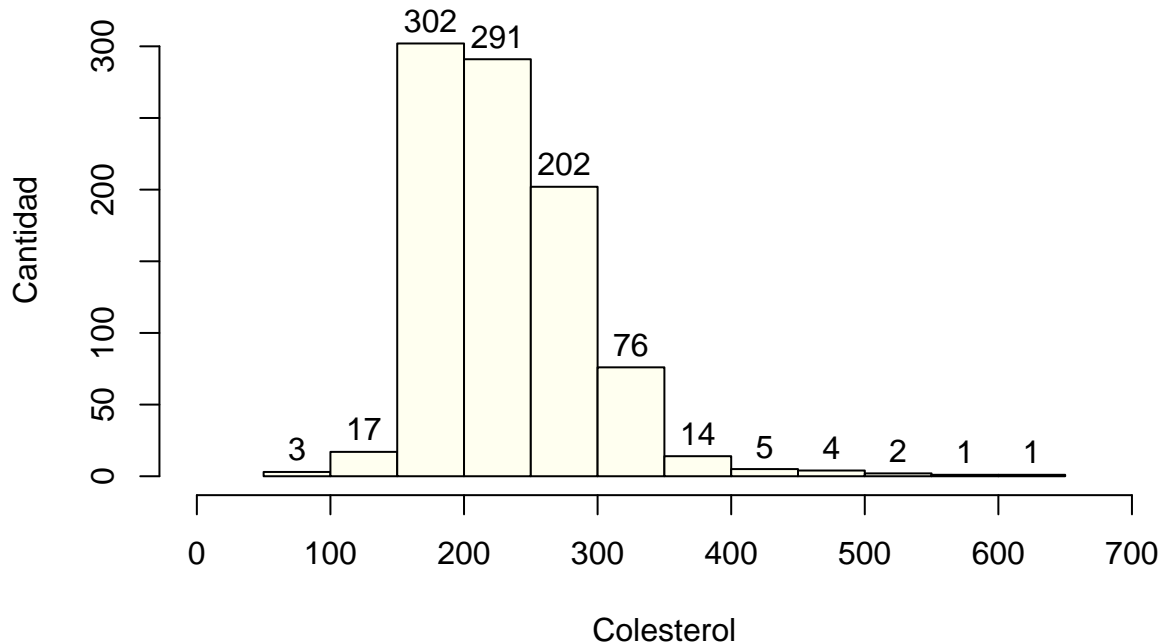
#vemos las estadísticas del dato
summary(datos$RestingBP)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      80.0   120.0   130.0   132.5   140.0   200.0
```

Ahora ya tenemos los valores entre 80 y 200 que son un rango normal para estos valores.

```
h1 <- hist(datos$Cholesterol, xlab="Colesterol", col="ivory",
           ylab="Cantidad", main="COLESTEROL", ylim = c(0,330), xlim = c(0, 700))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```

COLESTEROL

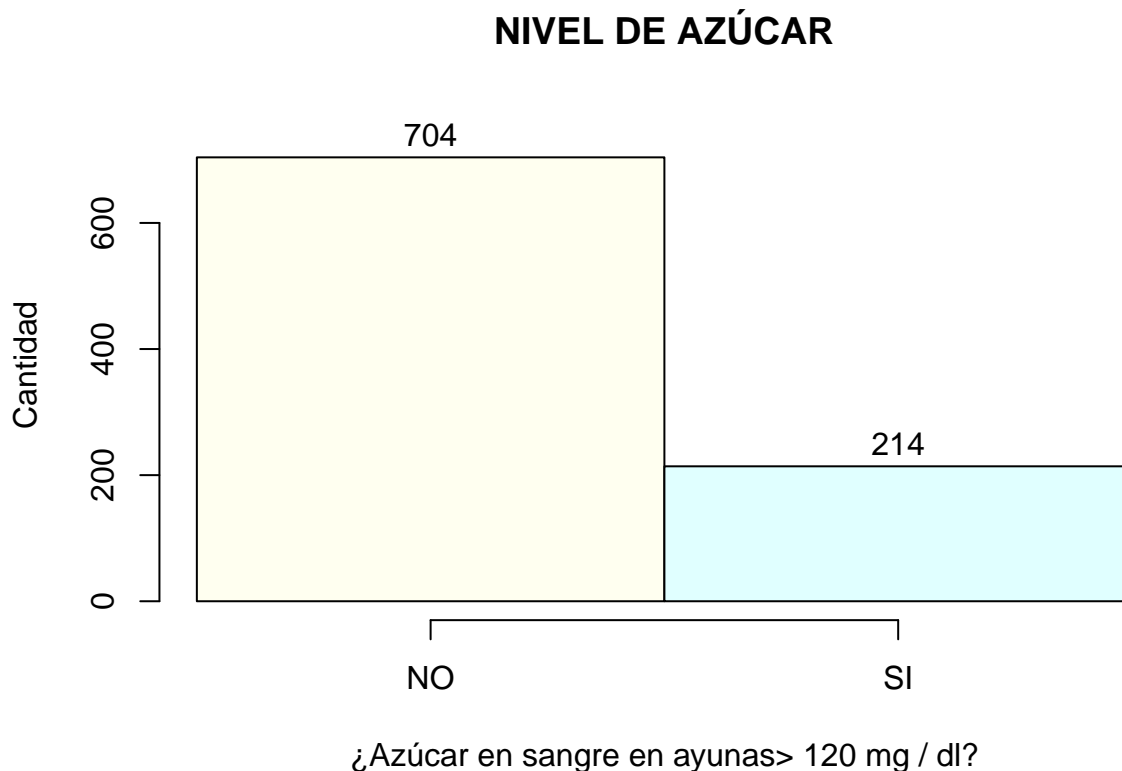


NIVEL DE AZÚCAR EN SANGRE EN AYUNAS (FastingBS)

Como se puede comprobar el conjunto de los datos pueden ser 1 o 0, es decir verdadero o falso si se cumple la siguiente condición: si nivel de azúcar en sangre en ayunas > 120 mg / dl.

En esta característica no tenemos valores nulos, así que vamos a ver la distribución de las dos opciones:

```
h1 <- hist(datos$FastingBS, xlab="¿Azúcar en sangre en ayunas> 120 mg / dl?",
           col=c("ivory", "lightcyan"), ylab="Cantidad",
           main="NIVEL DE AZÚCAR", breaks = 2, ylim = c(0, 750), axes = FALSE)
text(h1$mids, h1$counts, labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75), cex.axis=1, labels = c("NO", "SI" ))
axis(2)
```



Como se puede comprobar que hay mas casos que NO se cumple esa condición de que SÍ.

ECG EN REPOSO (RestingECG)

Nos damos cuenta de que el conjunto de datos viene identificado por 3 variables categóricas: + Normal: Normal, + ST: con anomalía de la onda ST-T + LVH: que muestra una hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes. Normalizamos para tenerlo de tipo numérico todas las variables:

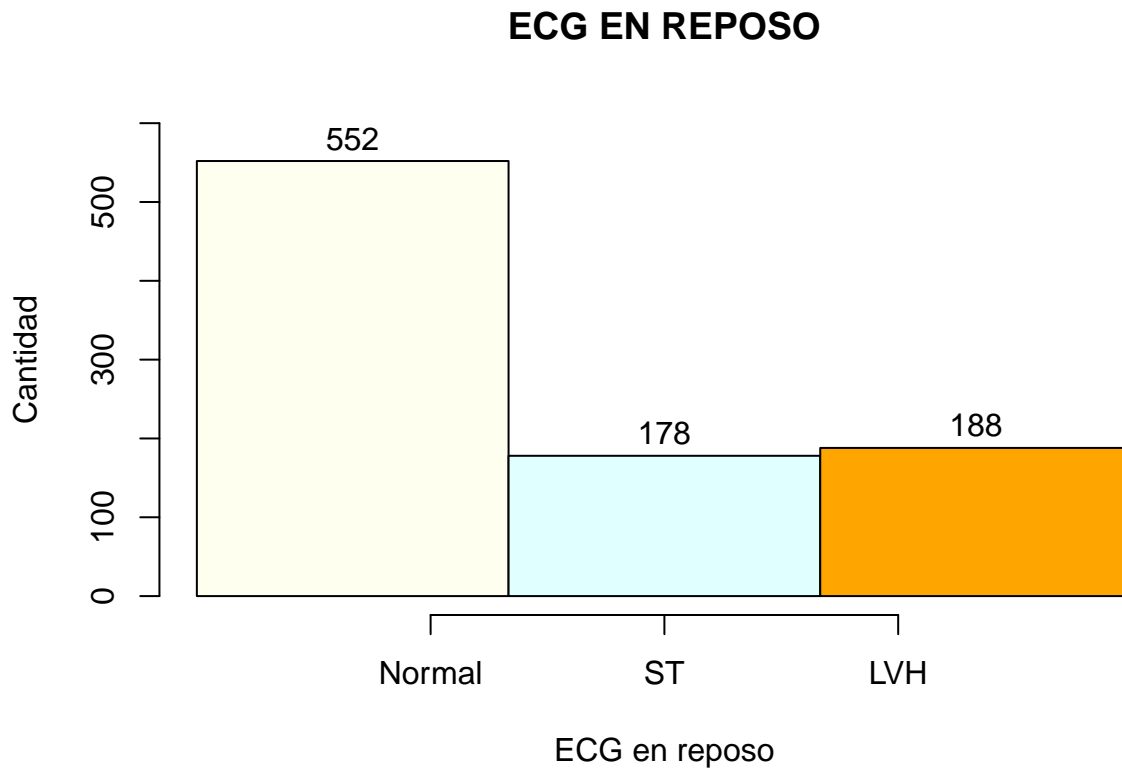
```
#Cambiamos las letras por los números
datos$RestingECG [datos$RestingECG == "Normal"] <- 0
datos$RestingECG [datos$RestingECG == "ST"] <- 1
datos$RestingECG [datos$RestingECG == "LVH"] <- 2

#Pasamos de carácter a numérico
datos$RestingECG <- as.numeric(datos$RestingECG)
```

Una vez normalizada la característica , analizamos el conjunto de los datos contemplados en esta.

```
h1 <- hist(datos$RestingECG, xlab="ECG en reposo",
           col= c("ivory", "lightcyan", "ORANGE"),
           ylab="Cantidad", main="ECG EN REPOSO",
           ylim = c(0, 600), axes = FALSE,
           breaks=seq(min(datos$RestingECG)-0.5,
                      max(datos$RestingECG)+0.5, by=1) )
```

```
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 1, 1.75 ), cex.axis=1, labels = c("Normal","ST", "LVH"))
axis(2)
```

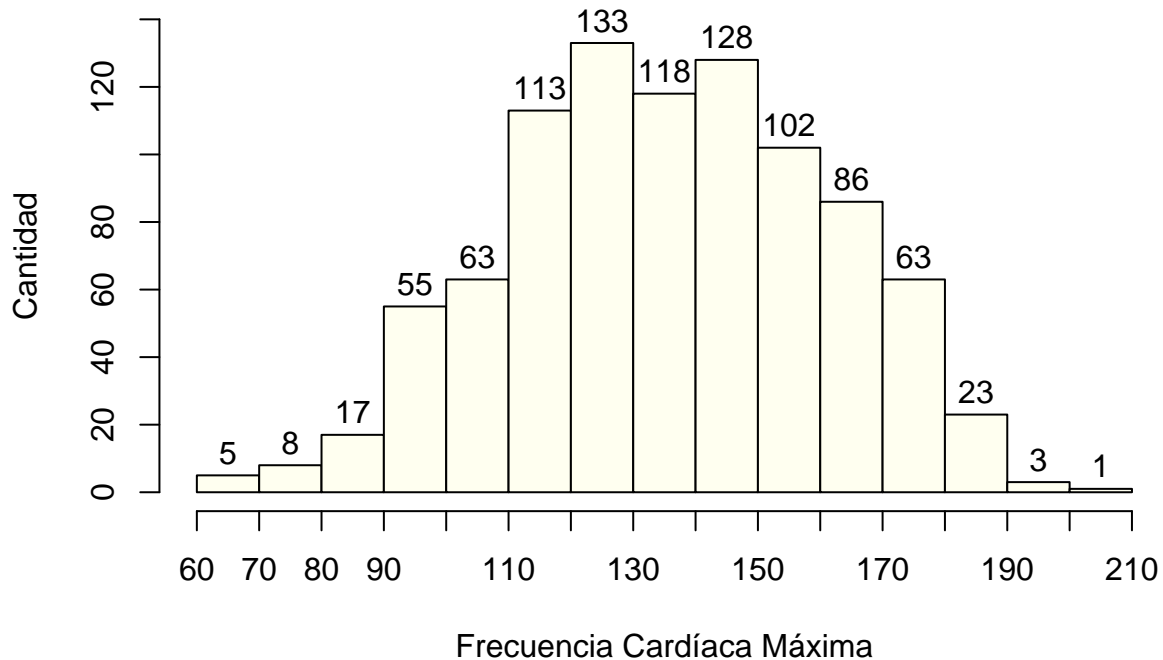


FRECUENCIA CARDÍACA MÁXIMA (MaxHR)

Dicha característica es de carácter numérica y en el conjunto de datos contempla valores desde el 60 al 202

```
h1 <- hist(datos$MaxHR, xlab="Frecuencia Cardíaca Máxima",
           col="ivory", ylab="Cantidad", main="FRECUENCIA CARDÍACA MÁXIMA",
           ylim = c(0,140), axes = FALSE)
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(60, 70, 80,90,100,110,120,130,140,150,160,170,180,190,200,210), cex.axis=1)
axis(2)
```

FRECUENCIA CARDÍACA MÁXIMA



Se puede comprobar que los extremos en el conjunto de datos tienen menos valores, y que el grueso de las muestras se encuentran entre los valores centrales (desde 100 a 180).

ANGINA INDUCIDA POR EJERCICIO (ExerciseAngina)

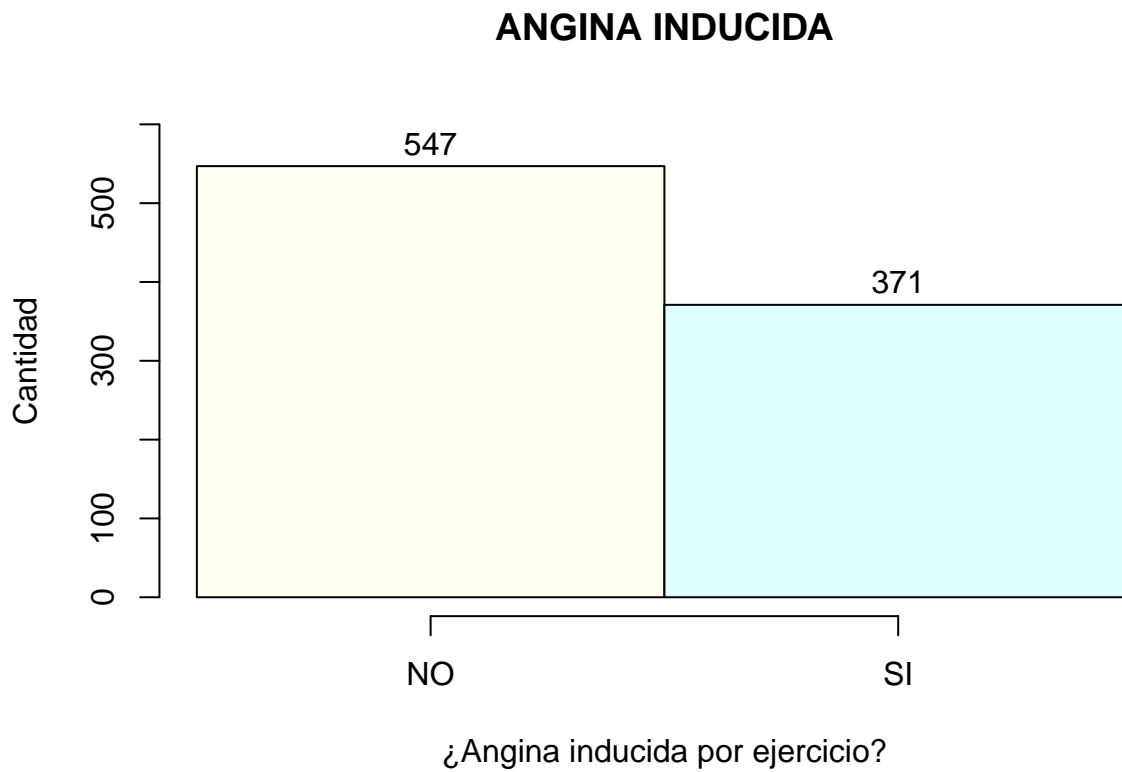
En el conjunto de datos tiene los valores Y: Sí, N: No. Al igual que se ha hecho con otras características, se normalizará el conjunto.

```
#Cambiamos las letras por los números
datos$ExerciseAngina [datos$ExerciseAngina == "N"] <- 0
datos$ExerciseAngina [datos$ExerciseAngina == "Y"] <- 1

#Pasamos de carácter a numérico
datos$ExerciseAngina <- as.numeric(datos$ExerciseAngina)
```

Una vez normalizada la característica , analizamos el conjunto de los datos contemplados en esta.

```
h1 <- hist(datos$ExerciseAngina, xlab="¿Angina inducida por ejercicio?",
           col=c("ivory", "lightcyan"), ylab="Cantidad", main="ANGINA INDUCIDA",
           breaks = 2, ylim = c(0, 600), axes = FALSE)
text(h1$mids, h1$counts, labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75), cex.axis=1, labels = c("NO", "SI" ))
axis(2)
```

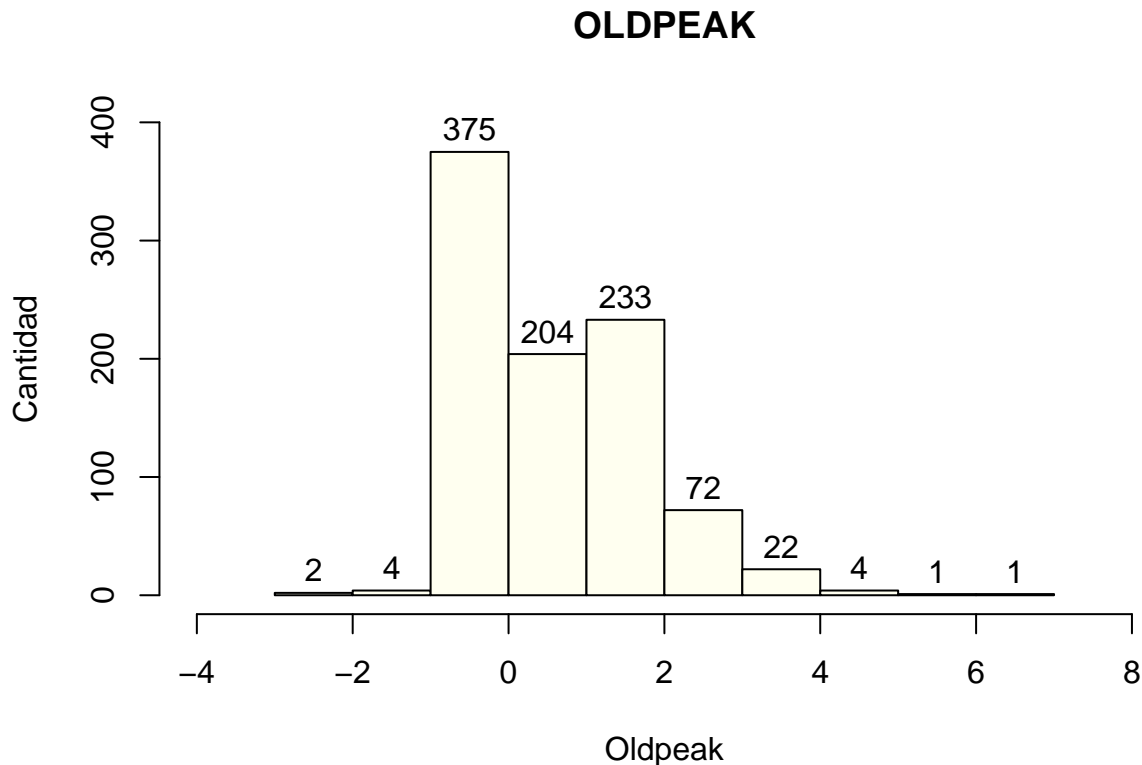


Como se puede apreciar, hay mas casos en que NO se ha producido una angina inducida por el ejercicio de que Si se haya producido.

OLDPEAK

Esta característica de tipo numérica puede abarcar valores negativos hasta hasta un máximo de un valor igual a 6,2.

```
h1 <- hist(datos$Oldpeak, xlab="Oldpeak", col="ivory", ylab="Cantidad", main="OLDPEAK", ylim = c(0,400))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```



Se puede comprobar que el grueso de las muestras se encuentra entre los valores centrales teniendo una distribución normal

PENDIENTE DEL SEGMENTO ST (ST_Slope)

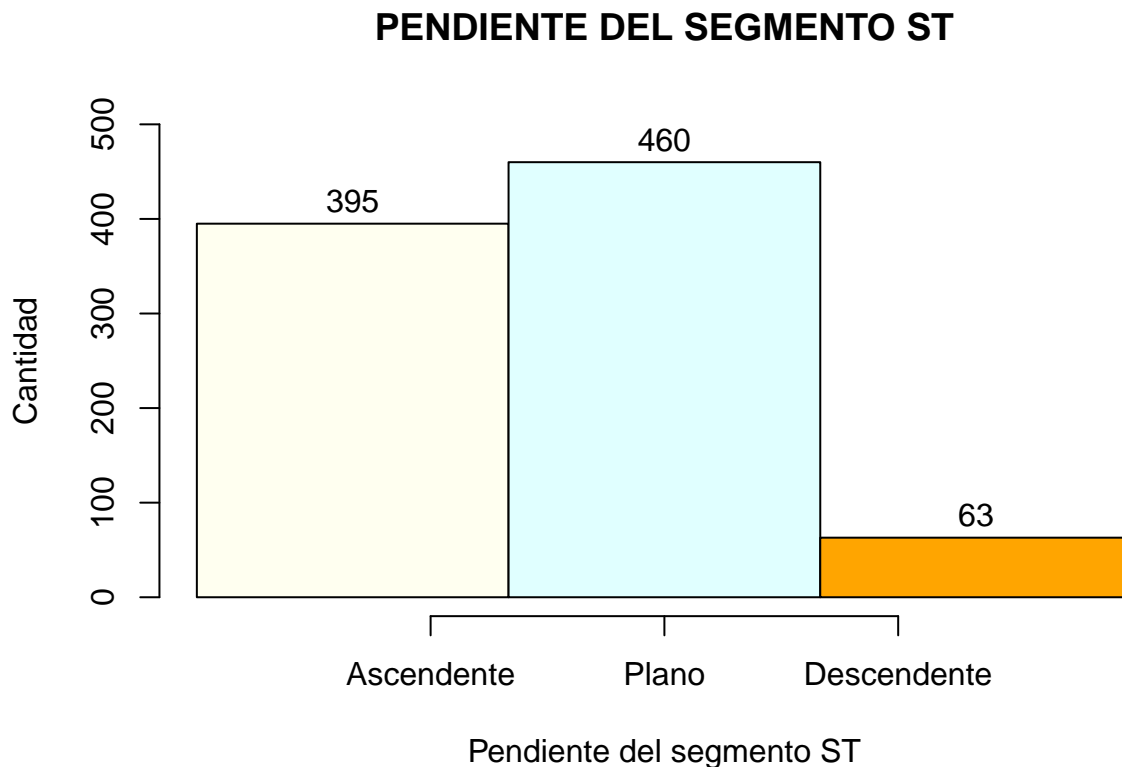
Como ocurría en otras características anteriores el conjunto tiene los valores para esta característica de la siguiente forma: + Up: uploping + Flat: flat + Down: downsloping Y como se ha realizado antes, se normalizará para solo tener datos numericos.

```
#Cambiamos las letras por los números
datos$ST_Slope [datos$ST_Slope == "Up"]    <- 0
datos$ST_Slope [datos$ST_Slope == "Flat"]  <- 1
datos$ST_Slope [datos$ST_Slope == "Down"]  <- 2

#Pasamos de carácter a numérico
datos$ST_Slope <- as.numeric(datos$ST_Slope)
```

Una vez normalizada la característica , analizamos el conjunto de los datos contemplados en esta.

```
h1 <- hist(datos$ST_Slope, xlab="Pendiente del segmento ST",
           col= c("ivory", "lightcyan", "ORANGE"), ylab="Cantidad",
           main="PENDIENTE DEL SEGMENTO ST", ylim = c(0, 500),
           axes = FALSE,breaks=seq(min(datos$ST_Slope)-0.5, max(datos$ST_Slope)+0.5, by=1) )
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25,1,1.75), cex.axis=1, labels = c("Ascendente","Plano", "Descendente"))
axis(2)
```

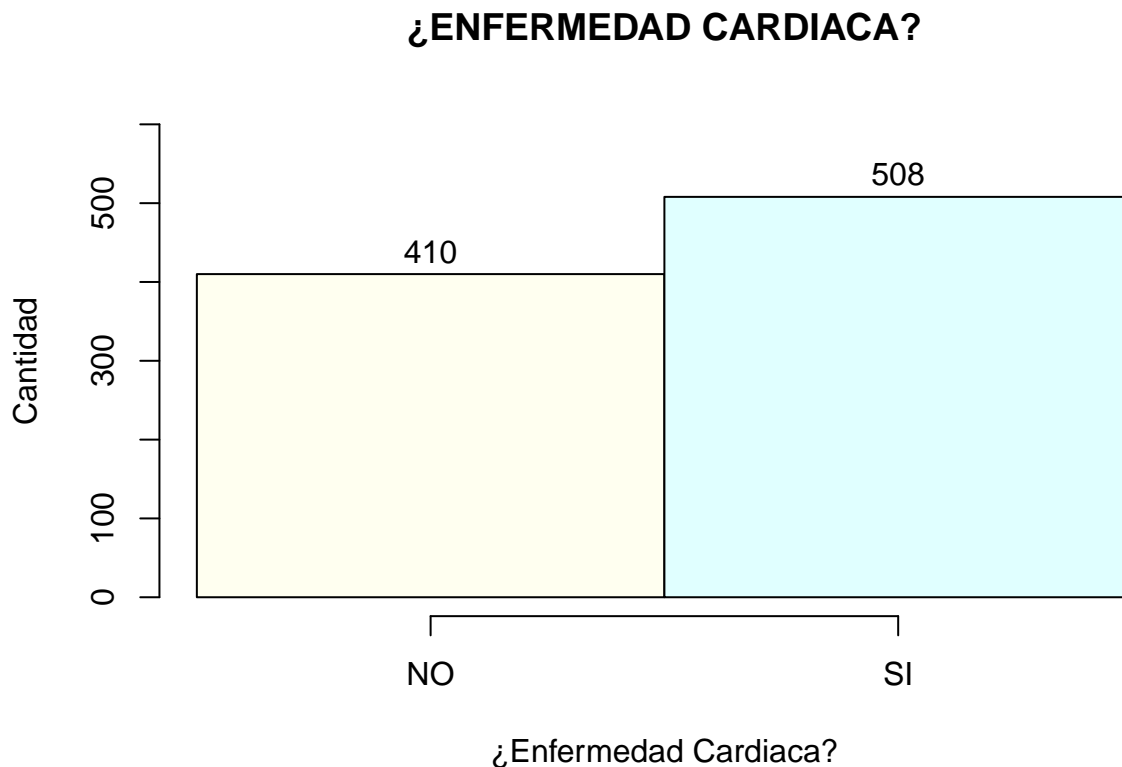



El caso más común es que la pendiente sea plana, teniendo menos casos en los casos descendentes.

¿ENFERMEDAD CARDIACA? (HeartDisease)

En el conjunto de datos tienen normalizada la salida usando el valor 1: enfermedad cardíaca, y el valor 0: Normal.

```
h1 <- hist(datos$HeartDisease, xlab="¿Enfermedad Cardiaca?",
           col=c("ivory", "lightcyan"),
           ylab="Cantidad", main="¿ENFERMEDAD CARDIACA?",
           breaks = 2, ylim = c(0, 600), axes = FALSE)
text(h1$mids, h1$counts, labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75), cex.axis=1, labels = c("NO", "SI" ))
axis(2)
```



Como se puede observar hay mas casos en que SI hay enfermedad cardiaca que caso en los que NO hay.

Construcción de conjunto de datos final

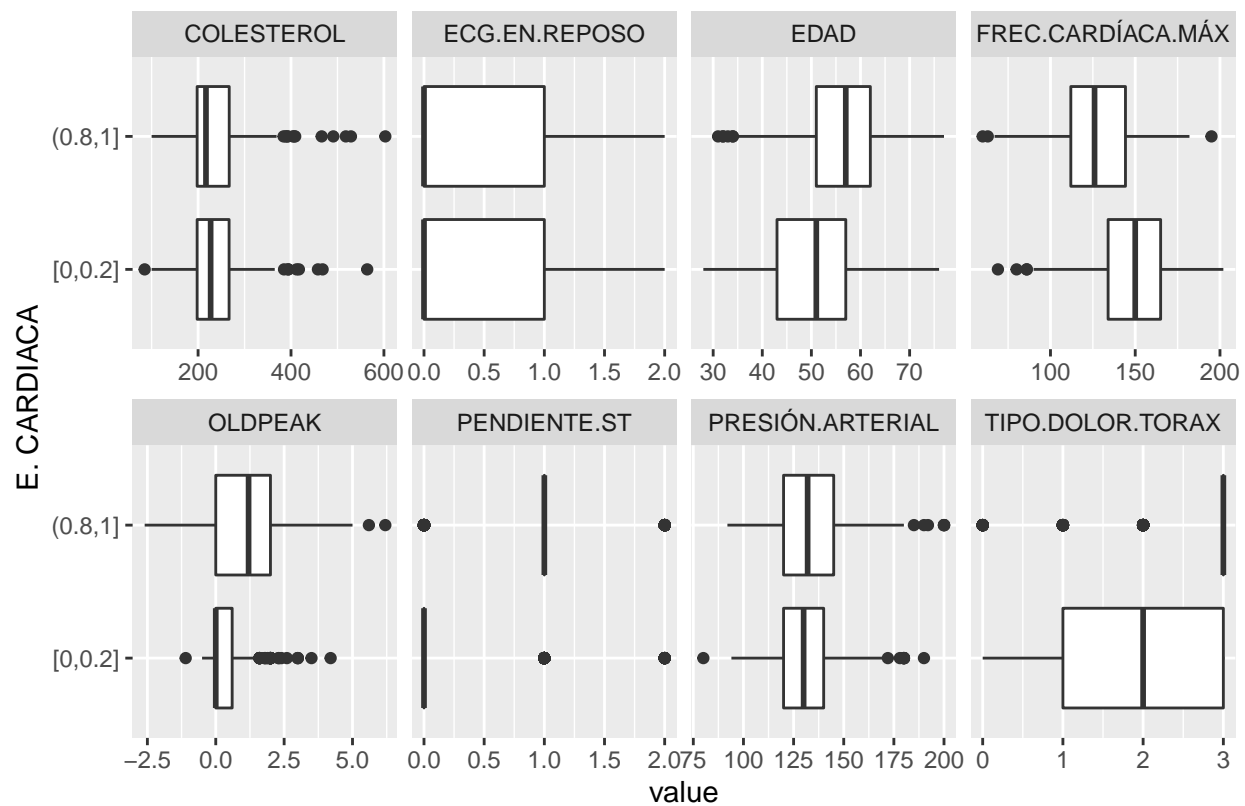
Renombramos las columnas para que tenga uno mas significativo y creamos el conjunto final de datos.

```
datos_final <- datos

colnames(datos_final)[1]<- "EDAD"
colnames(datos_final)[2]<- "SEXO"
colnames(datos_final)[3]<- "TIPO DOLOR TORAX"
colnames(datos_final)[4]<- "PRESIÓN ARTERIAL"
colnames(datos_final)[5]<- "COLESTEROL"
colnames(datos_final)[6]<- "NIVEL DE AZÚCAR"
colnames(datos_final)[7]<- "ECG EN REPOSO"
colnames(datos_final)[8]<- "FREC CARDÍACA MÁX"
colnames(datos_final)[9]<- "ANGINA x EJERCICIO"
colnames(datos_final)[10]<- "OLDPEAK"
colnames(datos_final)[11]<- "PENDIENTE ST"
colnames(datos_final)[12]<- "E. CARDIACA"
```

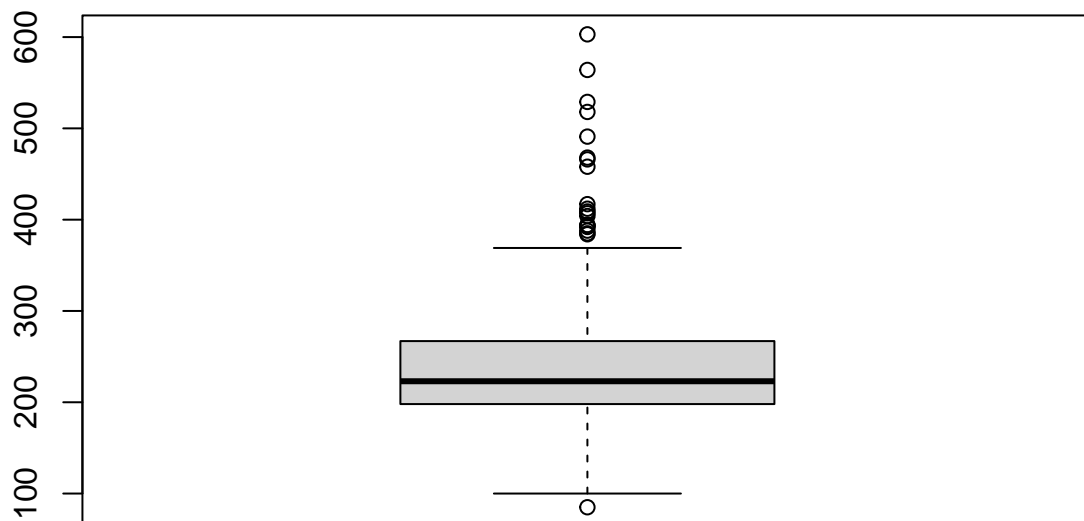
Por ultimo se va a mirar a través de los diagramas de cajas el rango de las características enfrenteado a si un paciente tiene una enfermedad cardiaca o no.

```
#Diagrama de caja de todas las características enfrentadas a si un paciente tiene enfermedad cardiaca
plot_boxplot(datos_final, by = "E. CARDIACA")
```



```
## Eliminamos outliers
```

```
datos_bp.cholesterol <- boxplot(datos_final$COLESTEROL)
```



```
datos_bp.cholesterol.out <- datos_bp.cholesterol$out
print("Eliminamos Outliers de la variable COLESTEROL con valores: ")
```

```
## [1] "Eliminamos Outliers de la variable COLESTEROL con valores: "
```

```
datos_bp.cholesterol.out
```

```
## [1] 468 518 412 529 85 392 466 393 388 603 404 491 394 458 384 385 564 407 417
## [20] 409 394
```

```
datos_final <- datos_final %>% filter(!(COLESTEROL %in% datos_bp.cholesterol.out))
dev.off()
```

```
## null device
##          1
```

```
datos_freq cardiaca.max <- boxplot(datos_final$`FREC CARDÍACA MÁX`)
datos_freq cardiaca.max.out <- datos_freq cardiaca.max$out
print("Eliminamos Outliers de la variable FREC CARDIACA MAX con valores: ")
```

```
## [1] "Eliminamos Outliers de la variable FREC CARDIACA MAX con valores: "
```

```
datos_frec.cardiaca.max.out
```

```
## [1] 63 60
```

```
datos_final <- datos_final %>% filter(!(`FREC CARDÍACA MÁX` %in% datos_frec.cardiaca.max.out))
dev.off()
```

```
## null device
##          1
```

```
datos_oldpeak <- boxplot(datos_final$OLDPEAK)
datos_oldpeak.out <- datos_oldpeak$out
print("Eliminamos Outliers de la variable OLDPEAK con valores: ")
```

```
## [1] "Eliminamos Outliers de la variable OLDPEAK con valores: "
```

```
datos_oldpeak.out
```

```
## [1] 4.0 5.0 -2.6 4.0 4.0 4.0 4.0 4.2 4.0 5.6 3.8 4.2 6.2 4.4 4.0
```

```
datos_final <- datos_final %>% filter(!(OLDPEAK %in% datos_oldpeak.out))
dev.off()
```

```
## null device
##          1
```

```
datos_bp.presion_arterial <- boxplot(datos_final$`PRESIÓN ARTERIAL`)
datos_bp.presion_arterial.out <- datos_bp.presion_arterial$out
print("Eliminamos Outliers de la variable PRESIÓN ARTERIAL ST con valores: ")
```

```
## [1] "Eliminamos Outliers de la variable PRESIÓN ARTERIAL ST con valores: "
```

```
datos_bp.presion_arterial.out
```

```
## [1] 190 180 180 200 180 180 180 80 200 185 200 180 180 178 172 180 190 174 180
## [20] 192 178 180 180 172
```

```
datos_final <- datos_final %>% filter(!(`PRESIÓN ARTERIAL` %in% datos_bp.presion_arterial.out))
dev.off()
```

```
## null device
##          1
```

```
datos_bp.tipo_dolor_torax <- boxplot(datos_final$`TIPO DOLOR TORAX`)
datos_bp.tipo_dolor_torax.out <- datos_bp.tipo_dolor_torax$out
print("Eliminamos Outliers de la variable TIPO DOLOR TORAX con valores: ")
```

```
## [1] "Eliminamos Outliers de la variable TIPO DOLOR TORAX con valores: "
```

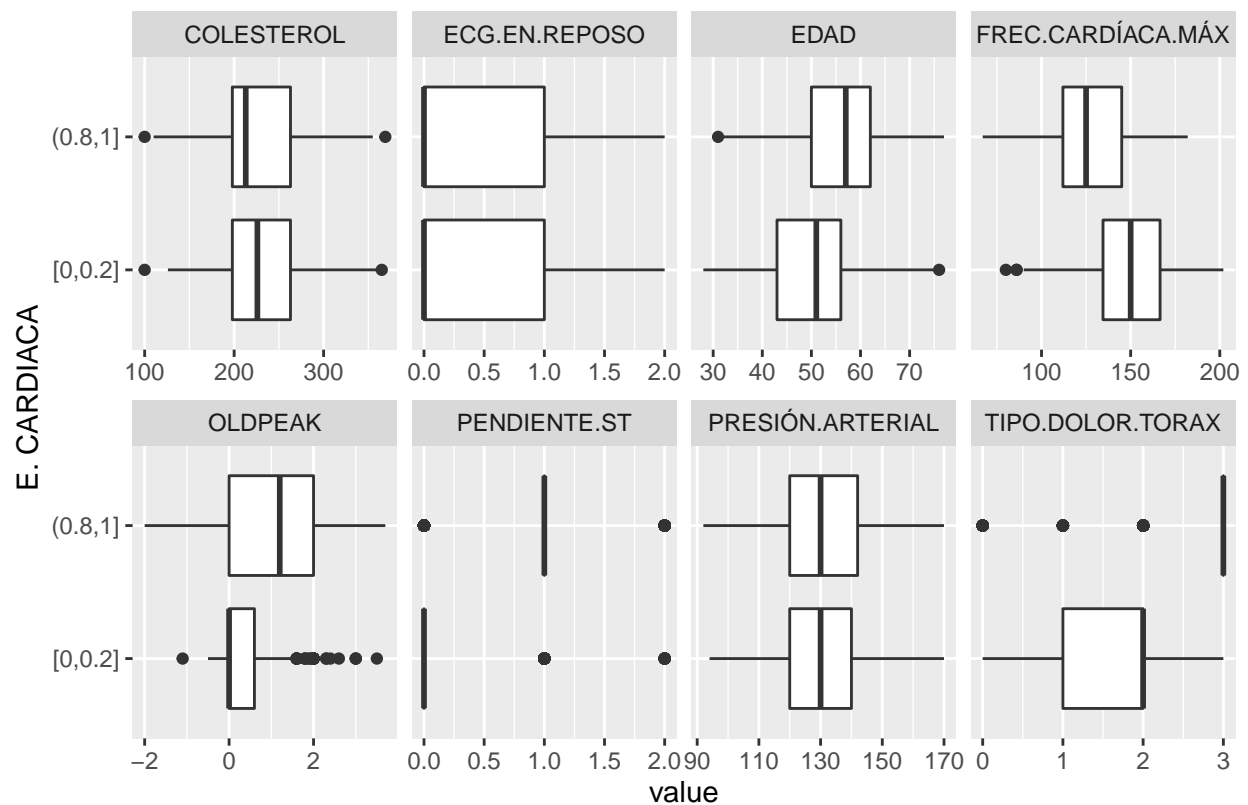
```
datos_bp.tipo_dolor_torax.out
```

```
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0
```

```
datos_final <- datos_final %>% filter(!(`PRESIÓN ARTERIAL` %in% datos_bp.tipo_dolor_torax.out))
dev.off()
```

```
## null device
##      1
```

```
#Diagrama de caja de todas las características enfrentadas a si un paciente tiene enfermedad cardiaca
plot_boxplot(datos_final, by = "E. CARDIACA")
```



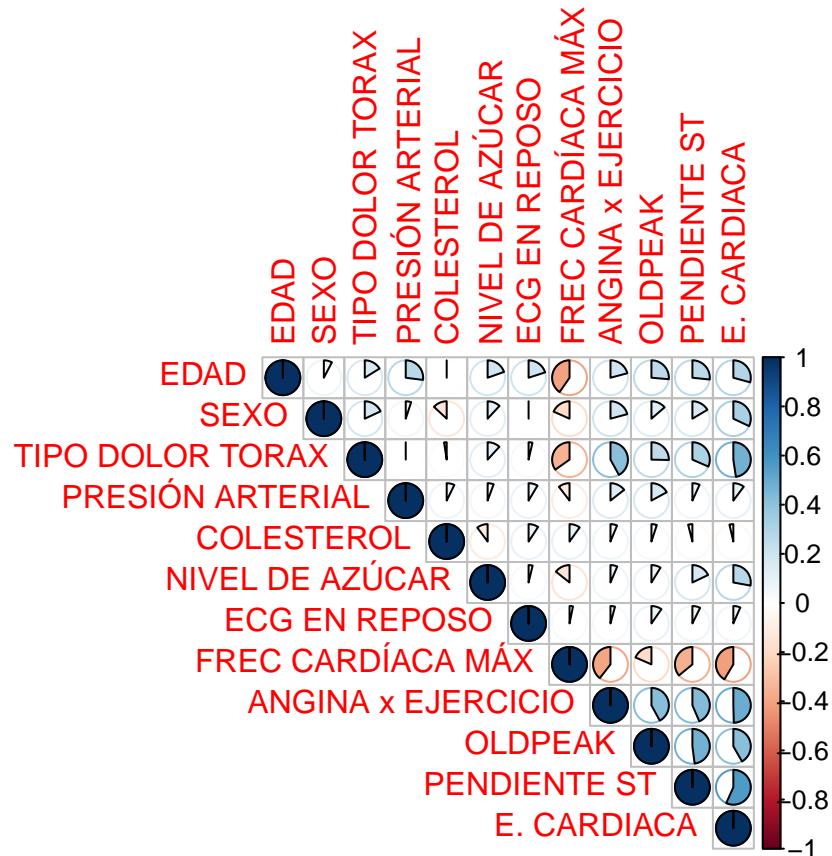
Correlaciones

```
#Calculamos las correlaciones
cor_datos <- cor(datos_final)
cor_datos
```

```
##          EDAD          SEXO TIPO DOLOR TORAX PRESIÓN ARTERIAL
```

## EDAD	1.00000000	0.076895936	0.16177251	0.27010020
## SEXO	0.07689594	1.000000000	0.18062187	0.05025163
## TIPO DOLOR TORAX	0.16177251	0.180621871	1.00000000	0.00575207
## PRESIÓN ARTERIAL	0.27010020	0.050251635	0.00575207	1.00000000
## COLESTEROL	0.00646818	-0.130868253	-0.02417120	0.07249940
## NIVEL DE AZÚCAR	0.19583798	0.118880787	0.11778422	0.05610760
## ECG EN REPOSO	0.20016219	0.000214852	0.03557102	0.08533045
## FREC CARDÍACA MÁX	-0.40368651	-0.183386990	-0.34773204	-0.10449449
## ANGINA x EJERCICIO	0.20858078	0.208003601	0.41992082	0.14412384
## OLDPEAK	0.26354926	0.136488498	0.25323659	0.17037057
## PENDIENTE ST	0.26643636	0.164908377	0.31548567	0.06775693
## E. CARDIACA	0.29274572	0.320210874	0.47674720	0.10195885
##	COLESTEROL	NIVEL DE AZÚCAR	ECG EN REPOSO	FREC CARDÍACA MÁX
## EDAD	0.00646818	0.19583798	0.200162193	-0.40368651
## SEXO	-0.13086825	0.11888079	0.000214852	-0.18338699
## TIPO DOLOR TORAX	-0.02417120	0.11778422	0.035571020	-0.34773204
## PRESIÓN ARTERIAL	0.07249940	0.05610760	0.085330450	-0.10449449
## COLESTEROL	1.00000000	-0.10158988	0.091946101	0.09899353
## NIVEL DE AZÚCAR	-0.10158988	1.00000000	0.037517622	-0.14315145
## ECG EN REPOSO	0.09194610	0.03751762	1.000000000	0.03201197
## FREC CARDÍACA MÁX	0.09899353	-0.14315145	0.032011973	1.00000000
## ANGINA x EJERCICIO	0.06179115	0.06366221	0.041701085	-0.39490163
## OLDPEAK	0.04996820	0.08719085	0.097204750	-0.18469918
## PENDIENTE ST	-0.03944674	0.17914306	0.073265085	-0.35936367
## E. CARDIACA	-0.03612672	0.27849741	0.065687178	-0.41445930
##	ANGINA x EJERCICIO	OLDPEAK	PENDIENTE ST	E. CARDIACA
## EDAD	0.20858078	0.26354926	0.26643636	0.29274572
## SEXO	0.20800360	0.13648850	0.16490838	0.32021087
## TIPO DOLOR TORAX	0.41992082	0.25323659	0.31548567	0.47674720
## PRESIÓN ARTERIAL	0.14412384	0.17037057	0.06775693	0.10195885
## COLESTEROL	0.06179115	0.04996820	-0.03944674	-0.03612672
## NIVEL DE AZÚCAR	0.06366221	0.08719085	0.17914306	0.27849741
## ECG EN REPOSO	0.04170108	0.09720475	0.07326508	0.06568718
## FREC CARDÍACA MÁX	-0.39490163	-0.18469918	-0.35936367	-0.41445930
## ANGINA x EJERCICIO	1.00000000	0.42043058	0.43494651	0.49603806
## OLDPEAK	0.42043058	1.00000000	0.47956054	0.41712200
## PENDIENTE ST	0.43494651	0.47956054	1.00000000	0.56709534
## E. CARDIACA	0.49603806	0.41712200	0.56709534	1.00000000

```
#Representación de las correlaciones
corrplot(cor_datos, method = "pie", type="upper")
```



Análisis de componentes principales (PCA)

Ahora se va a realizar un análisis de componentes sobre el conjunto de datos final. Lo primero que vamos a calcular es la varianza de todas las características

```
#Cálculo de la varianza de los componentes.
var <- apply(datos_final, 2, var)
var
```

```
##          EDAD          SEXO  TIPO DOLOR TORAX  PRESIÓN ARTERIAL
##      90.3141048      0.1628341      0.8744589      250.5232279
##      COLESTEROL  NIVEL DE AZÚCAR      ECG EN REPOSO  FREC CARDÍACA MÁX
##      2138.3888670      0.1773829      0.6460062      636.7883465
## ANGINA x EJERCICIO      OLDPEAK      PENDIENTE ST      E. CARDIACA
##      0.2392250      0.9438371      0.3559272      0.2484219
```

Como se puede observar de una manera bastante clara, el colesterol es la característica que mas varia de un individuo a otro.

Lo siguiente es centrar y escalar las características, para que así las variables pierdan esa variabilidad. Una vez calculada la matriz se la asigno al pca

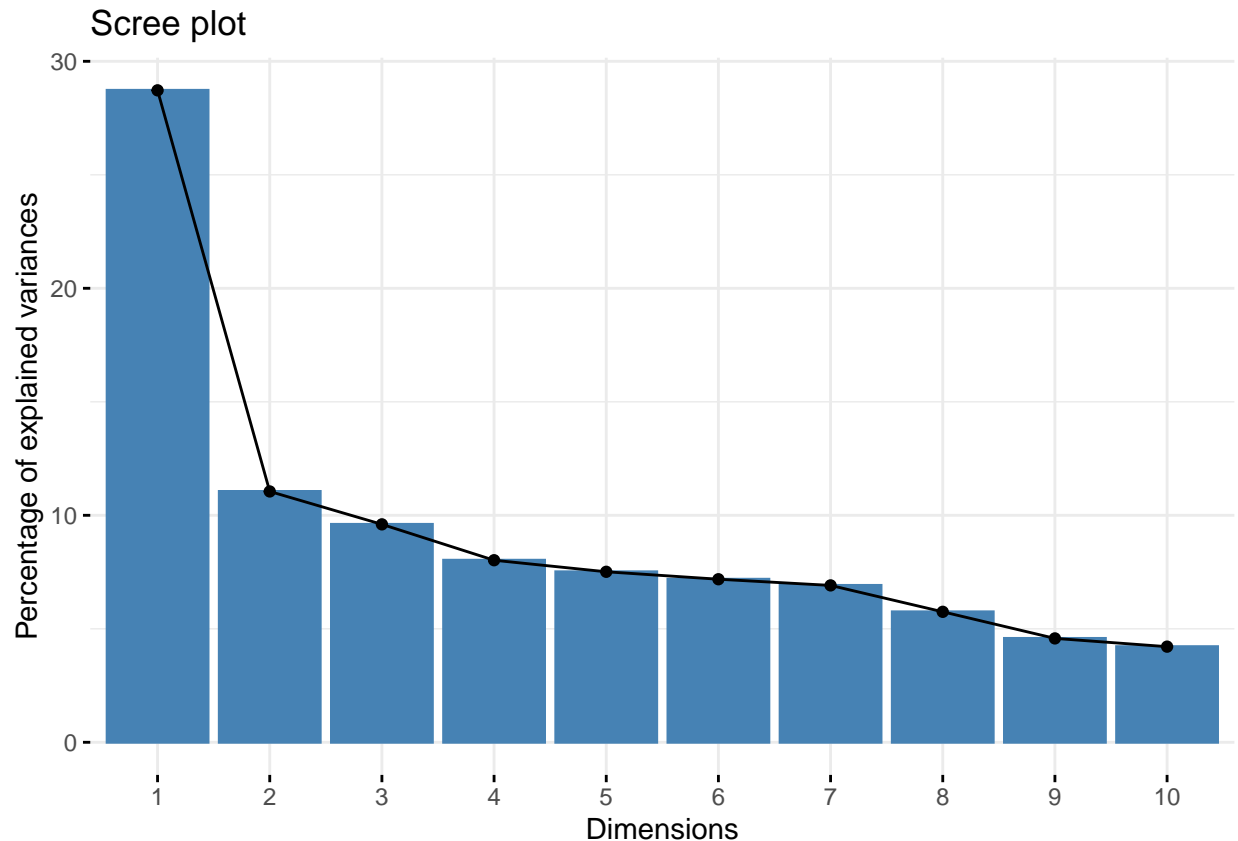
```
#Calculo de la descomposición de los componentes
pca <- prcomp(datos_final, scale = TRUE, center = TRUE)
pca
```



```
## Standard deviations (1, ..., p=12):
## [1] 1.8564679 1.1515284 1.0733796 0.9810575 0.9494047 0.9284618 0.9107163
## [8] 0.8305224 0.7410896 0.7110669 0.6442692 0.6003302
##
## Rotation (n x k) = (12 x 12):
##
##          PC1          PC2          PC3          PC4          PC5
## EDAD      0.28222922 -0.339554033  0.40514821 -0.15820750  0.24316201
## SEXO      0.20561304  0.302877950  0.13369146  0.05667367 -0.75072948
## TIPO DOLOR TORAX 0.32785083  0.186589247 -0.22039063  0.06678534  0.16471366
## PRESIÓN ARTERIAL 0.13138428 -0.465698922  0.27085862 -0.50081786 -0.38194411
## COLESTEROL -0.02415369 -0.493452913 -0.45274796  0.02799784  0.07717871
## NIVEL DE AZÚCAR  0.17356613  0.109066340  0.52754644  0.34355554  0.18351815
## ECG EN REPOSO  0.07295270 -0.473265859  0.14363050  0.64731232 -0.10676234
## FREC CARDÍACA MÁX -0.34237265 -0.113736611 -0.14685467  0.34904671 -0.33854363
## ANGINA x EJERCICIO 0.38134229 -0.005442403 -0.31505110 -0.12153806 -0.05032265
## OLDPEAK    0.33569745 -0.184467793 -0.22863429  0.07657927 -0.16926560
## PENDIENTE ST  0.39057184  0.026765692 -0.12918151  0.13807684  0.08165285
## E. CARDIACA  0.43521390  0.119141723 -0.04841931  0.13083364 -0.02774428
##
##          PC6          PC7          PC8          PC9
## EDAD      -0.19471655  0.03114419 -0.33720382 -0.446483847
## SEXO      -0.17555031 -0.30381293 -0.34273824 -0.103366050
## TIPO DOLOR TORAX -0.25141156 -0.27315963  0.56895910 -0.459575825
## PRESIÓN ARTERIAL 0.16348266 -0.02432652  0.44184210  0.150218186
## COLESTEROL  0.19531662 -0.62816863 -0.30531610 -0.006483788
## NIVEL DE AZÚCAR  0.55952600 -0.33816677  0.12495518  0.068890568
## ECG EN REPOSO  -0.47462917  0.07791309  0.13387388  0.218754326
## FREC CARDÍACA MÁX 0.33202141  0.11388062  0.22482741 -0.288249590
## ANGINA x EJERCICIO -0.05241167 -0.03582937  0.15228451  0.363103474
## OLDPEAK    0.31651774  0.43363912 -0.09167292 -0.429983629
## PENDIENTE ST  0.20717499  0.32735970 -0.20103703  0.311477808
## E. CARDIACA  0.09355359 -0.08388655  0.02394786  0.081941936
##
##          PC10          PC11          PC12
## EDAD      0.044464422 -0.43990654  0.09650745
## SEXO      0.002698998  0.04661736  0.16689353
## TIPO DOLOR TORAX -0.182009879  0.08957373  0.25226173
## PRESIÓN ARTERIAL -0.208010976  0.08184172  0.02689832
## COLESTEROL -0.096351957  0.08828867  0.03341747
## NIVEL DE AZÚCAR  0.237895123  0.10270323  0.11417929
## ECG EN REPOSO  0.041396939  0.14047495 -0.03301944
## FREC CARDÍACA MÁX -0.069543888 -0.58182795  0.10784688
## ANGINA x EJERCICIO 0.636097106 -0.39817082  0.12554845
## OLDPEAK    0.295048715  0.42287354 -0.13819556
## PENDIENTE ST  -0.499353224 -0.06197682  0.51342190
## E. CARDIACA  -0.328884032 -0.26792641 -0.75705747
```

Se puede ver que la primera componente tiene la mayor desviación estándar de todos los componentes. Para verlo de una manera mas clara, se va a representar de una manera grafica la salida anterior

```
#Representación PCA's anteriores
fviz_eig(pca)
```



Como se ha visto antes, tanto de una manera numérica como gráfica, el PC1 es el que mejor de todos con una diferencia notable. Si usamos la técnica del codo, deberíamos coger solamente las dos primeras componentes.

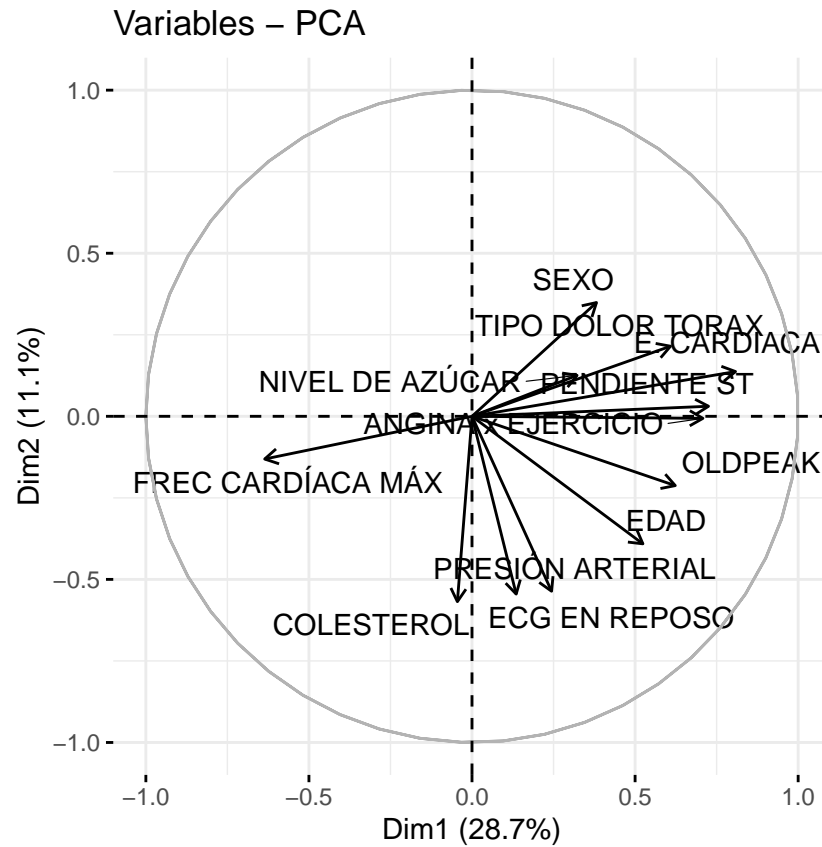
Para confirmar la interpretación, no estaría de más obtener las estadísticas de todas las componentes

```
#Estadísticas de las componentes
summary(pca)
```

```
## Importance of components:
##              PC1    PC2    PC3    PC4    PC5    PC6    PC7
## Standard deviation  1.8565 1.1515 1.07338 0.98106 0.94940 0.92846 0.91072
## Proportion of Variance 0.2872 0.1105 0.09601 0.08021 0.07511 0.07184 0.06912
## Cumulative Proportion 0.2872 0.3977 0.49372 0.57393 0.64904 0.72088 0.78999
##              PC8    PC9    PC10    PC11    PC12
## Standard deviation  0.83052 0.74109 0.71107 0.64427 0.60033
## Proportion of Variance 0.05748 0.04577 0.04213 0.03459 0.03003
## Cumulative Proportion 0.84747 0.89324 0.93538 0.96997 1.00000
```

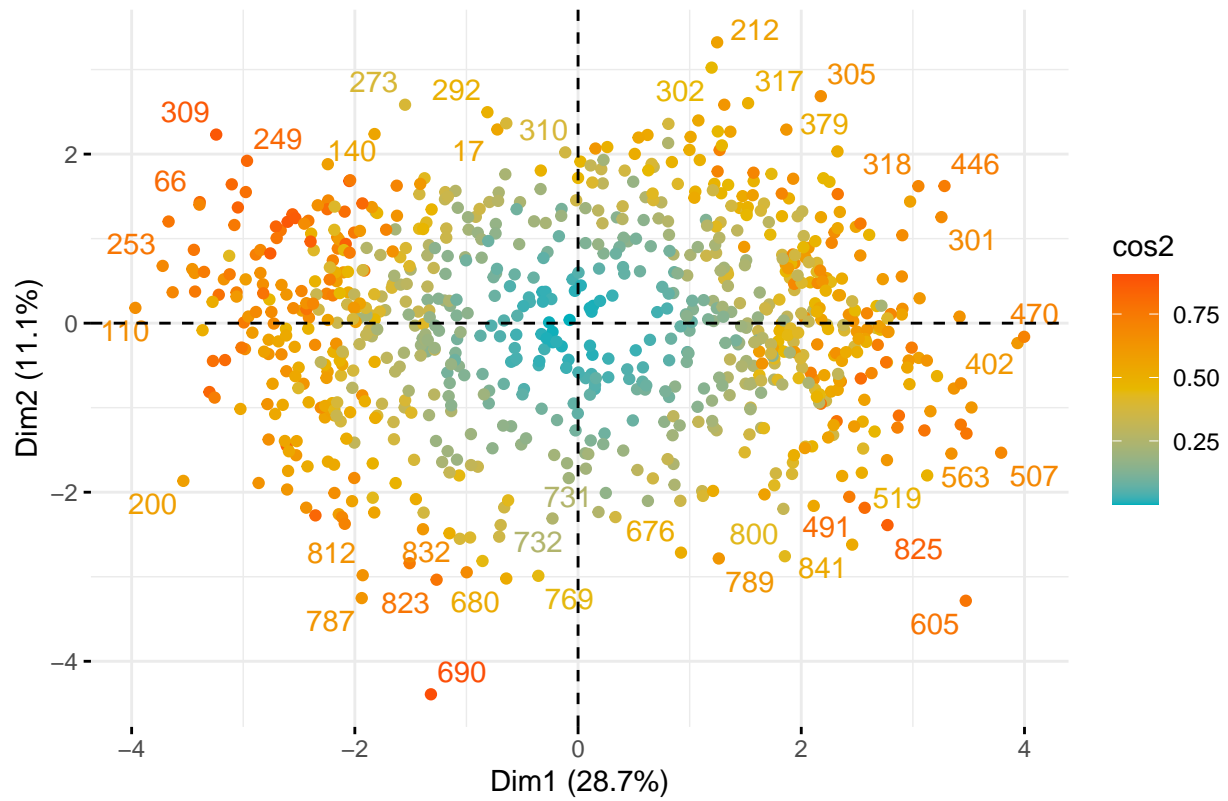
Viendo las estadísticas vemos que con las dos primeras componentes solamente podríamos explicar un 39,75% de los datos. Como no queremos perder información en el modelo, nos tendríamos que quedar con todas las componentes. Para verlo de una manera visual, se va a representar la PCA de una manera gráfica.

```
#Representación de variables sobre componentes principales
fviz_pca_var(pca, repel = TRUE, scale = 0)
```

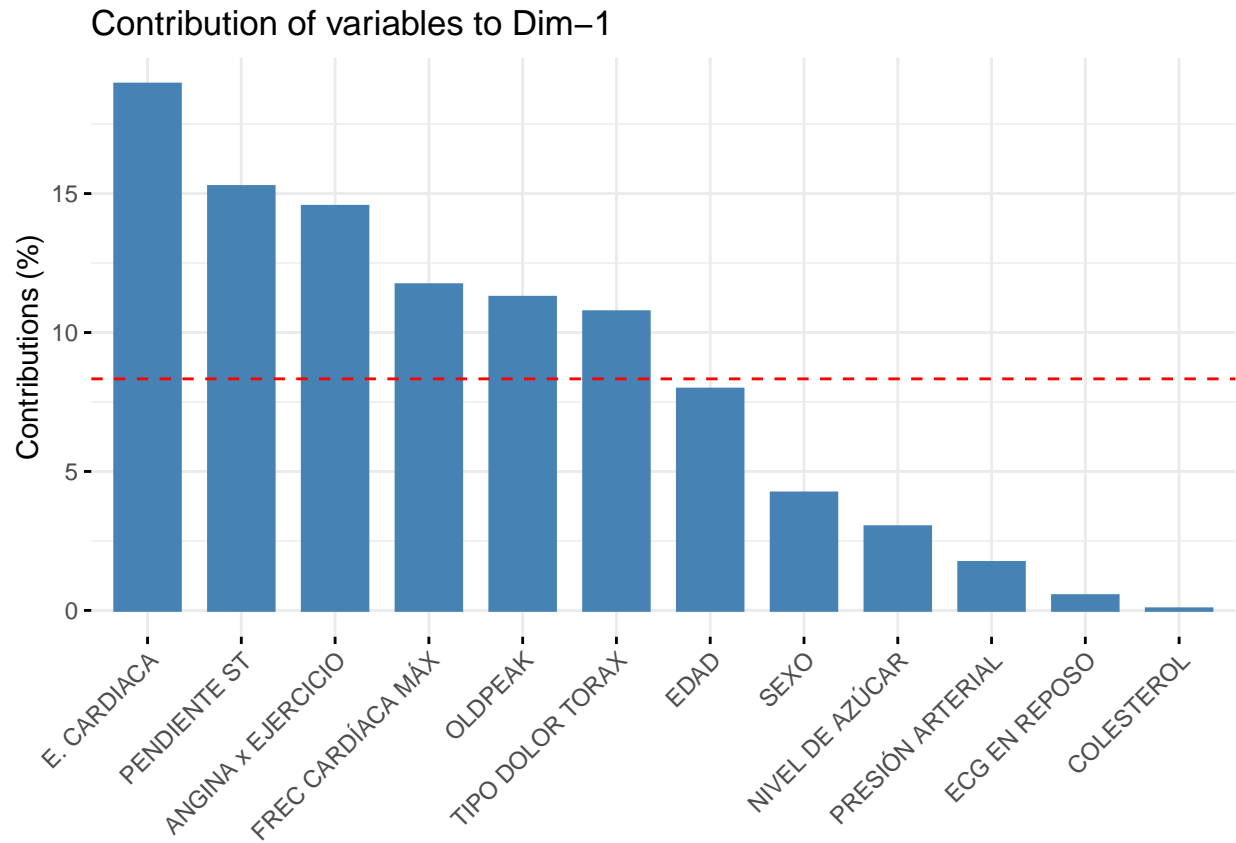


```
#Representación de observaciones sobre componentes principales
fviz_pca_ind(pca, col.ind = "cos2", gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRUE)
```

Individuals – PCA



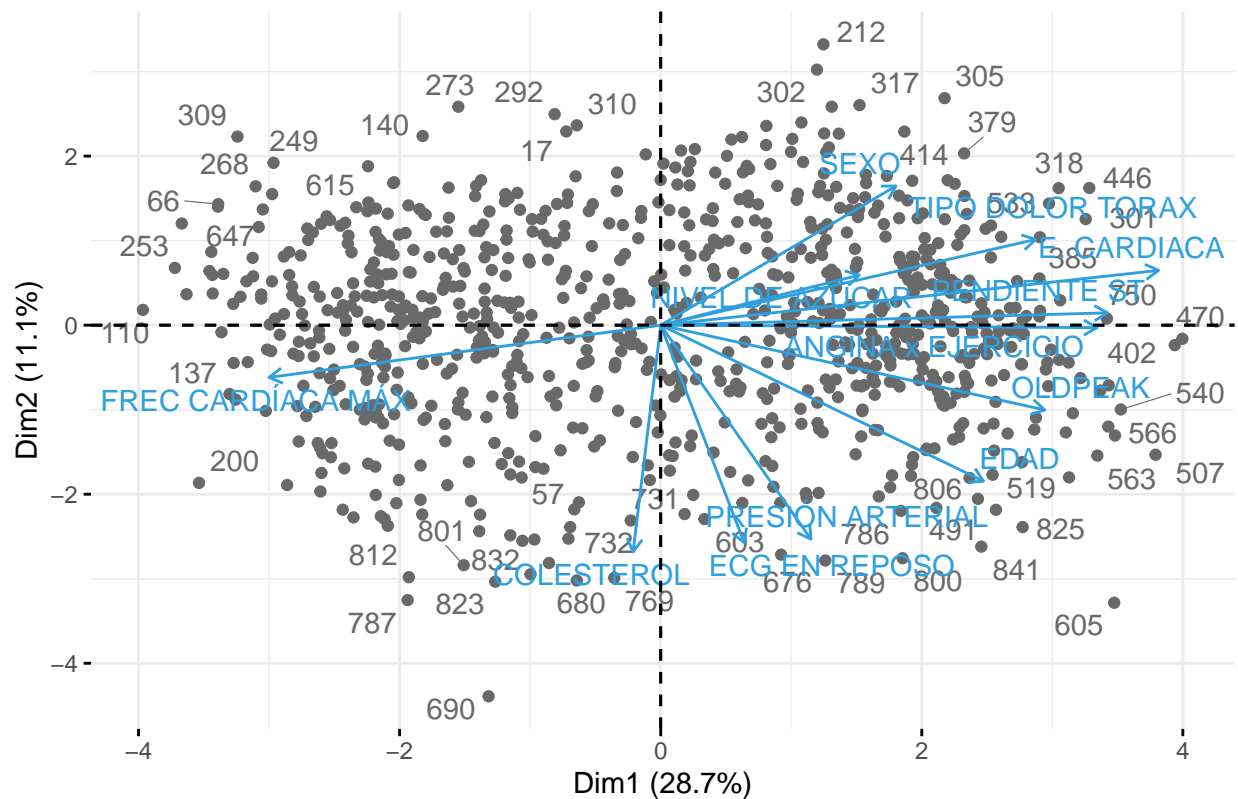
```
#Representa la contribución de filas/columnas de los resultados de un pca  
fviz_contrib(pca,choice = "var")
```



Una vez que hemos representada las variables y los individuos, se va a fusionar estas dos gráficas

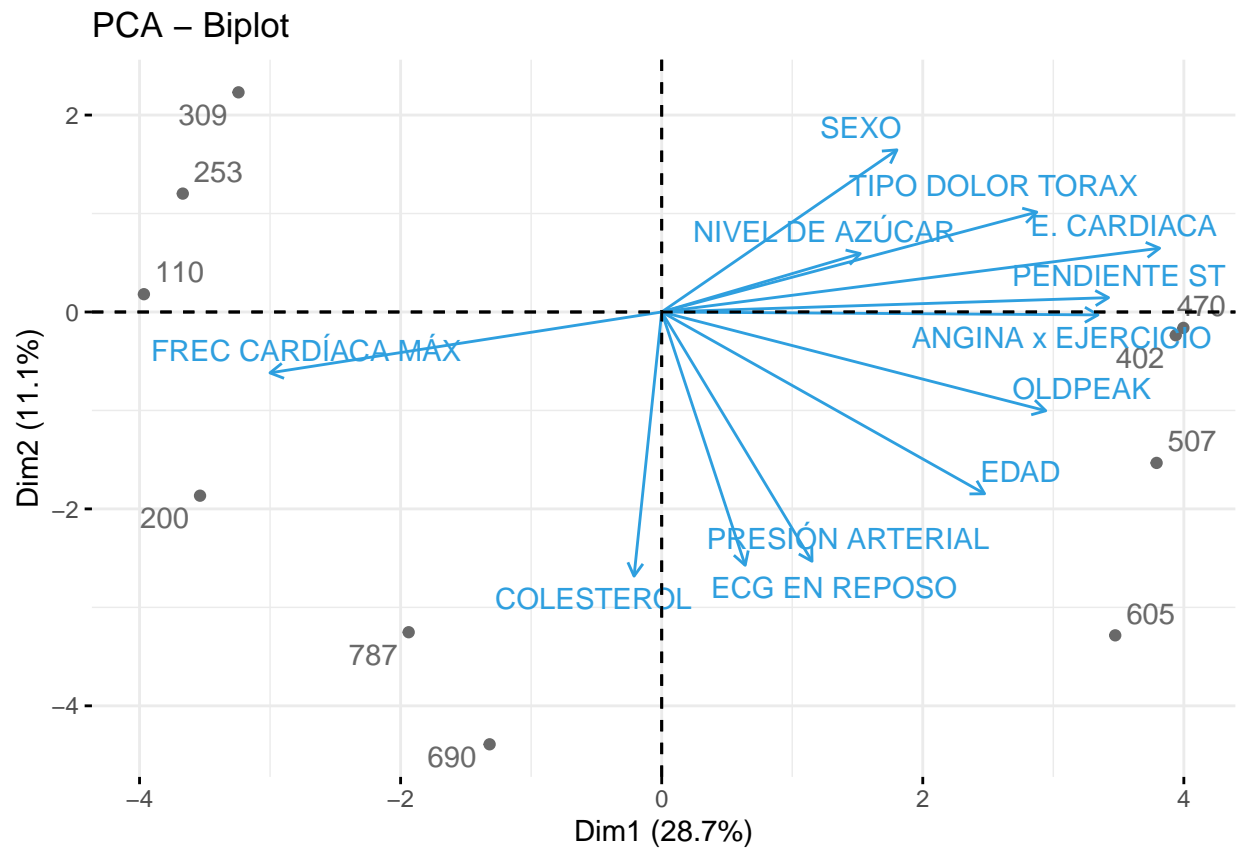
```
#Representación de variables y los individuos en la misma gráfica  
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF", col.ind = "#696969")
```

PCA – Biplot



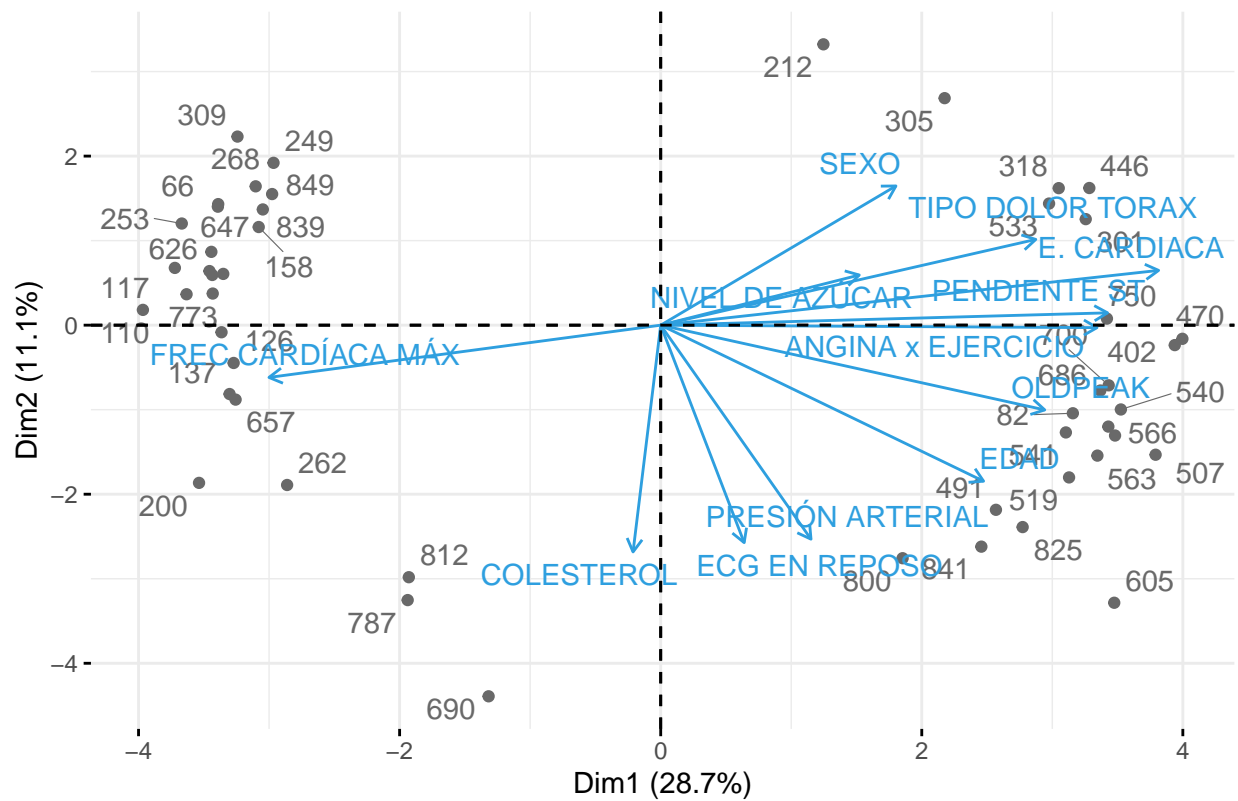
Aunque la opción de repelerse esta activada al ser bastantes casos no se puede ver una manera correcta, así que se a mostrar solamente los 10, 50 y 100 casos más influyentes

```
#Representación de variables y los 10 individuos más influyentes en la misma gráfica
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF",
  col.ind = "#696969", select.ind = list(contrib = 10))
```

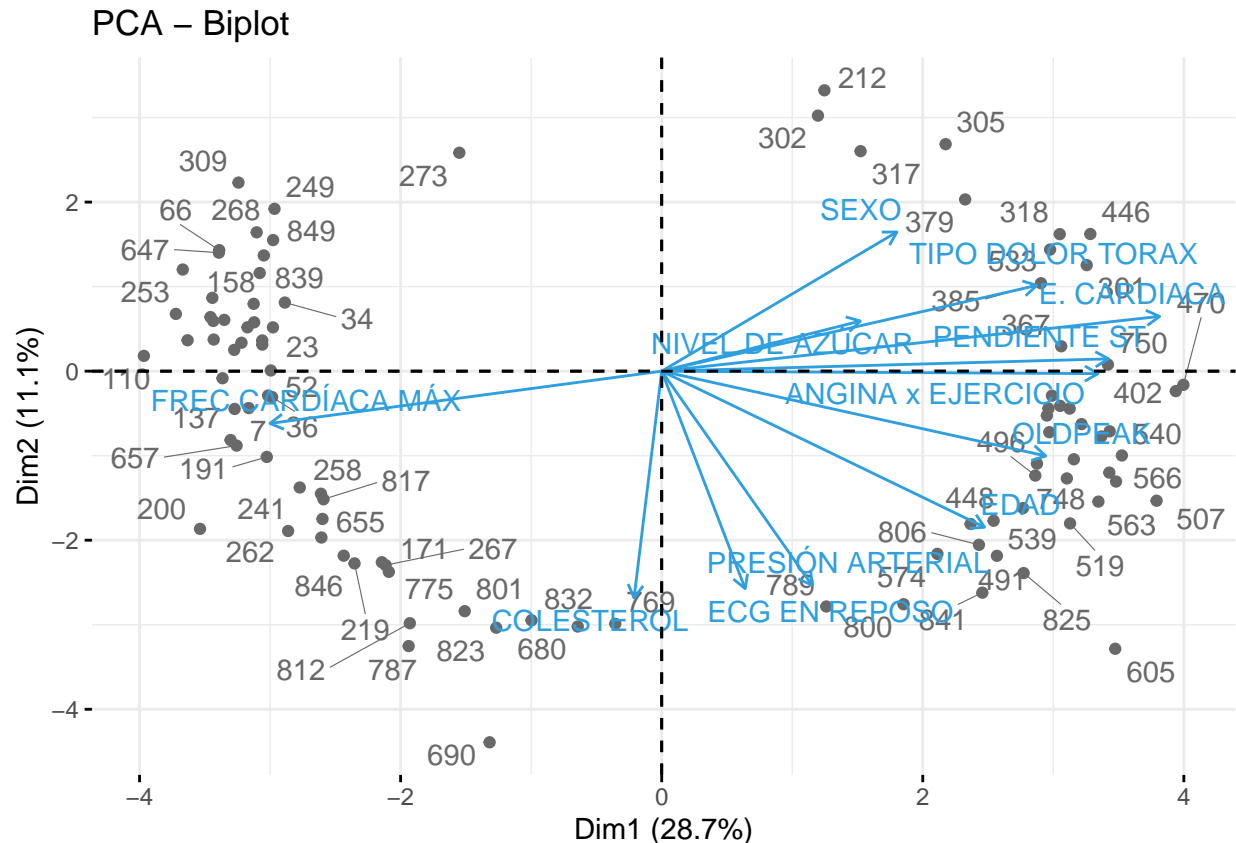


```
#Representación de variables y los 50 individuos más influyentes en la misma gráfica
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF",
  col.ind = "#696969", select.ind = list(contrib = 50))
```

PCA – Biplot



```
#Representación de variables y los 100 individuos más influyentes en la misma gráfica
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF",
  col.ind = "#696969", select.ind = list(contrib = 100))
```

Al mostrar solamente los casos mas influyentes, se puede ver con mas claridad las relaciones entre los individuos y las características. Podemos concluir de este análisis de componentes, que no se puede quitar ninguna característica ya que se perdería información.

Exportación de los datos

Una vez que hemos acometido sobre el conjunto de datos inicial los procedimientos de integración, validación y limpieza anteriores, procedemos a guardar estos en un nuevo fichero denominado heart_dissease_data_clean.csv:

```
# Exportación de los datos limpios en .csv
write.csv(datos_final, "./heart_dissease_data_clean.csv")
```

Análisis de los datos

COMprobación de la normalidad y la homogeneidad de la varianza

Para la comprobación de que los valores que toman nuestra variables cuantitativa provienen de una distribución normal vamos a utilizar la prueba de normalidad de Anderson-Darling.

Podemos comprobar que para cada prueba se obtiene un p-valor superior al nivel de significancia estadística prefijado en $\alpha = 0,05$. Si esto se cumple, entonces se considera que variable en cuestion sigue la distribución normal.

```

if (!require('nortest')) install.packages('nortest'); library('nortest')
alpha = 0.05
col.names = colnames(datos_final)
ind = 1

# Comprobanos unicamente 1S variables que inicialmente eran de tipo numericas

for (i in colnames(datos_final)) {
  if (ind == 1) cat("Variables que no siguen una distribución normal:\n")
  if(vector_tipos[ind] == "numeric")
  {
    p_val = ad.test(unlist(datos_final[i]))$p.value
    if (p_val < alpha) {
      cat(i)
      # Format output
      if (ind < ncol(datos) - 1) cat(", ")
      if (ind %% 3 == 0) cat("\n")
    }
  }
  ind = ind + 1
}

```

```

## Variables que no siguen una distribución normal:
## EDAD, PRESIÓN ARTERIAL, COLESTEROL, NIVEL DE AZÚCAR,
## FREC CARDÍACA MÁX, OLDPEAK, E. CARDIACA

```

Podemos realizar un Q-Q plot para comprobar si las variables obtenidas en el anterior punto no siguen una distribución normal.

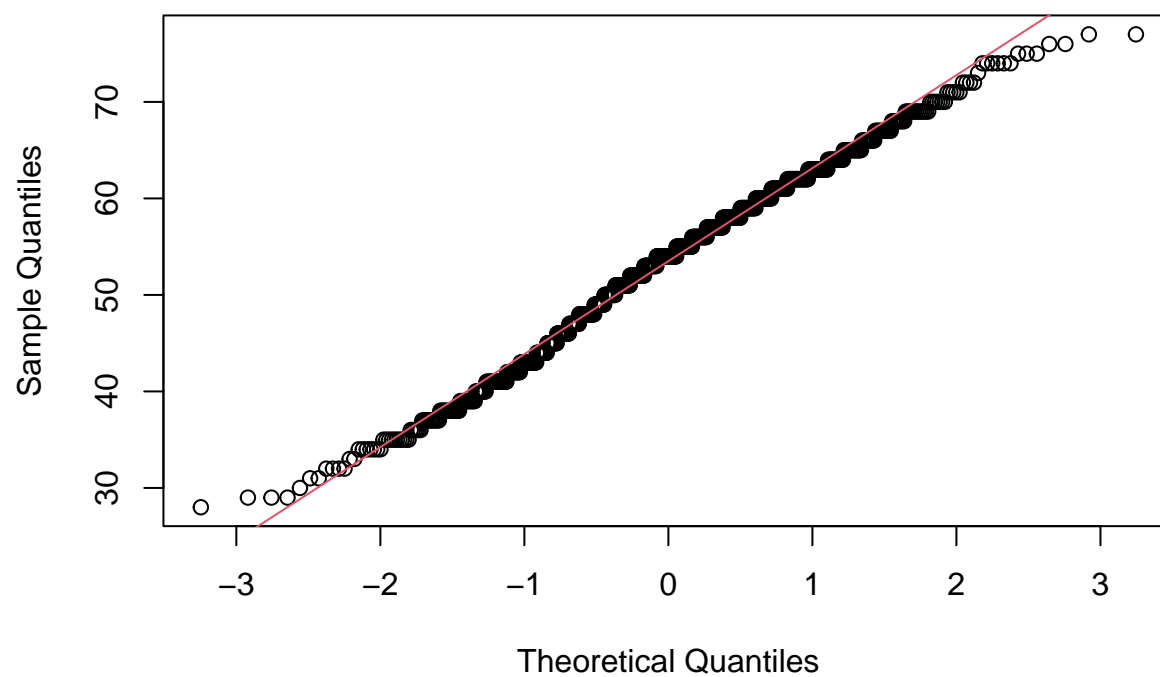
```

# Edad
variables <- c("EDAD", "PRESIÓN ARTERIAL", "COLESTEROL",
               "FREC CARDÍACA MÁX", "OLDPEAK")

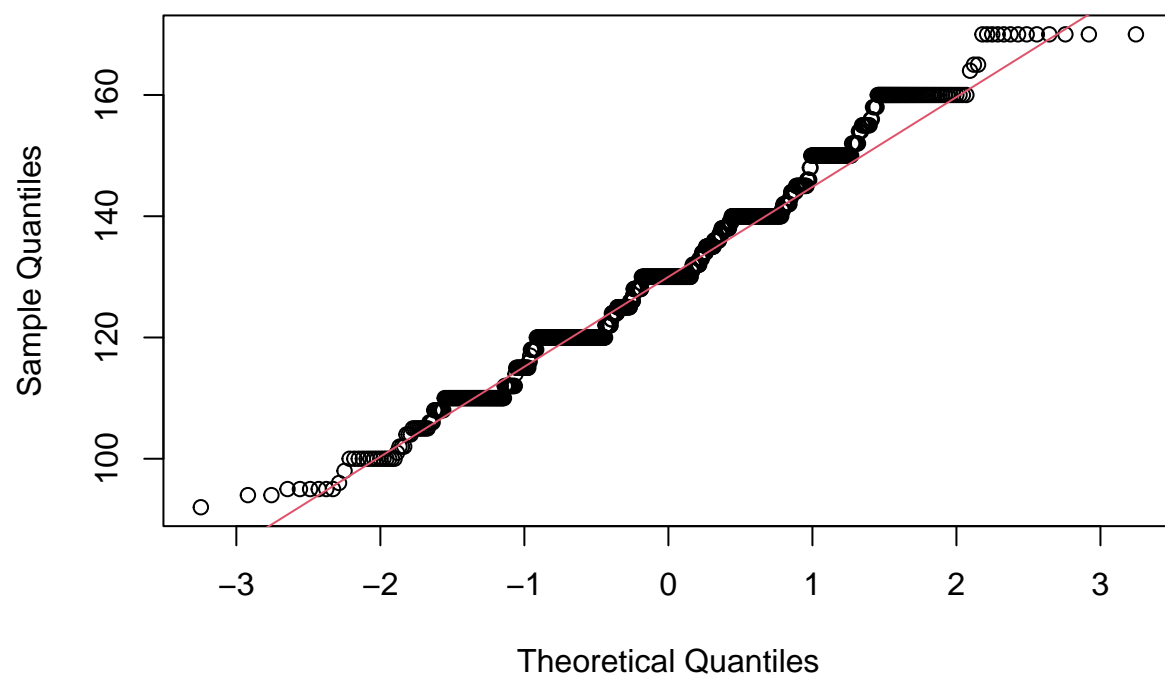
for(i in(variables))
{
  qqnorm(unlist(datos_final[i]), main = paste0("Q-Q para la variable: ", i));qqline(unlist(datos_final
}

```

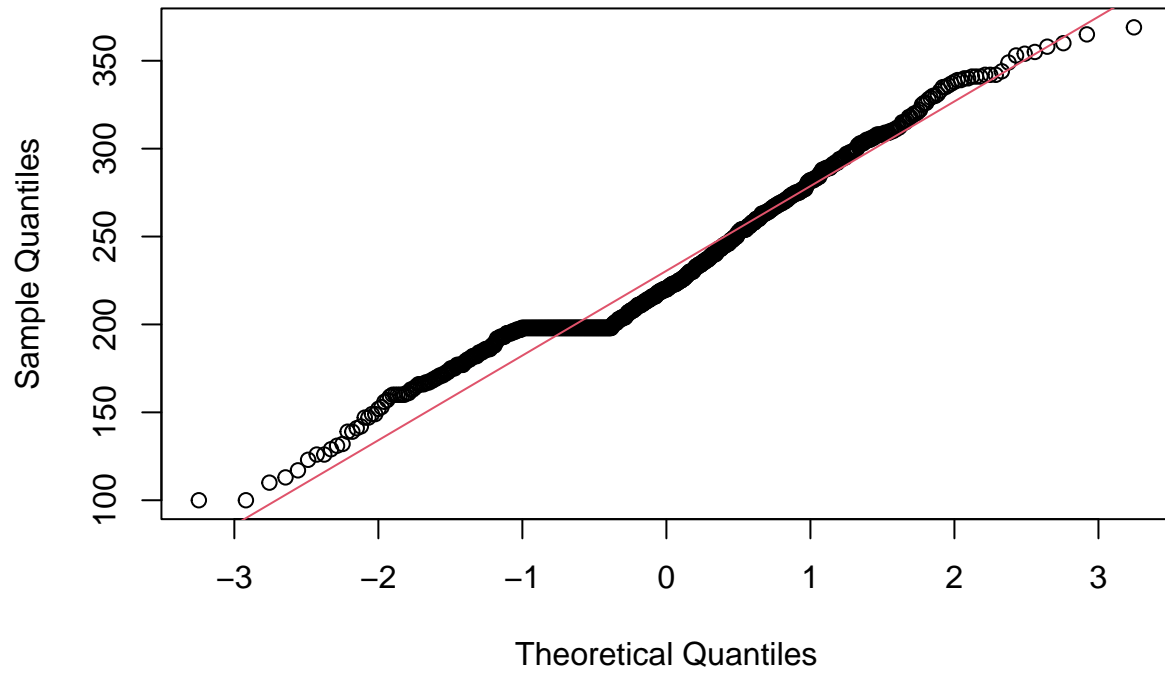
Q-Q para la variable: EDAD



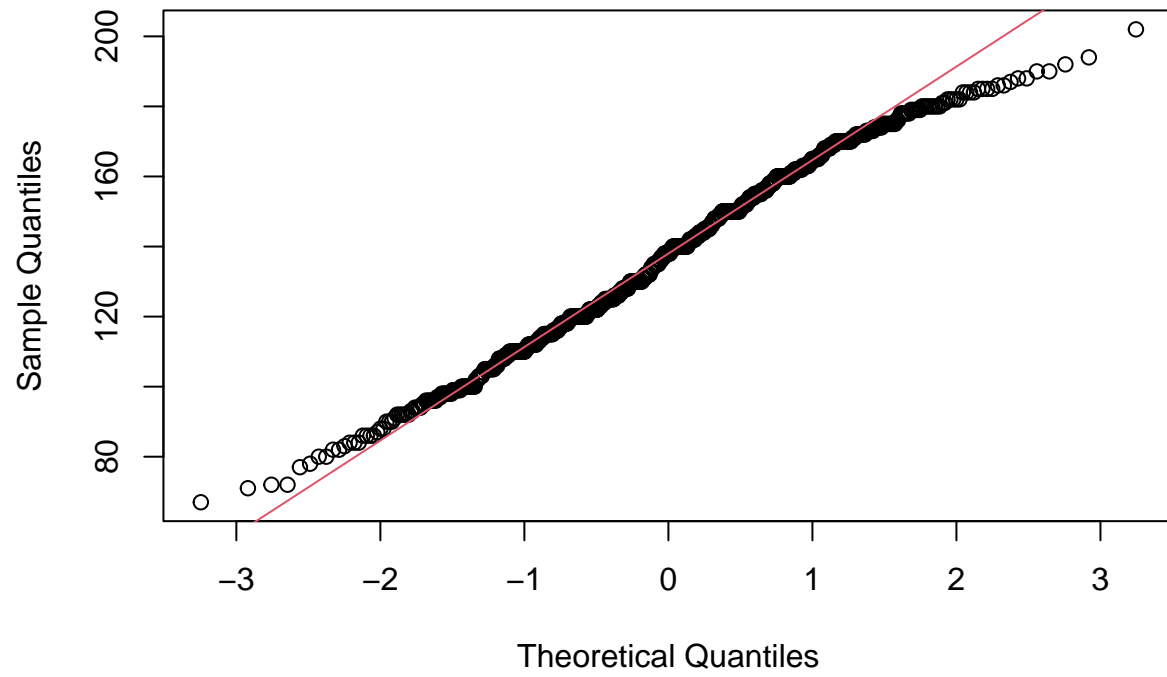
Q-Q para la variable: PRESIÓN ARTERIAL



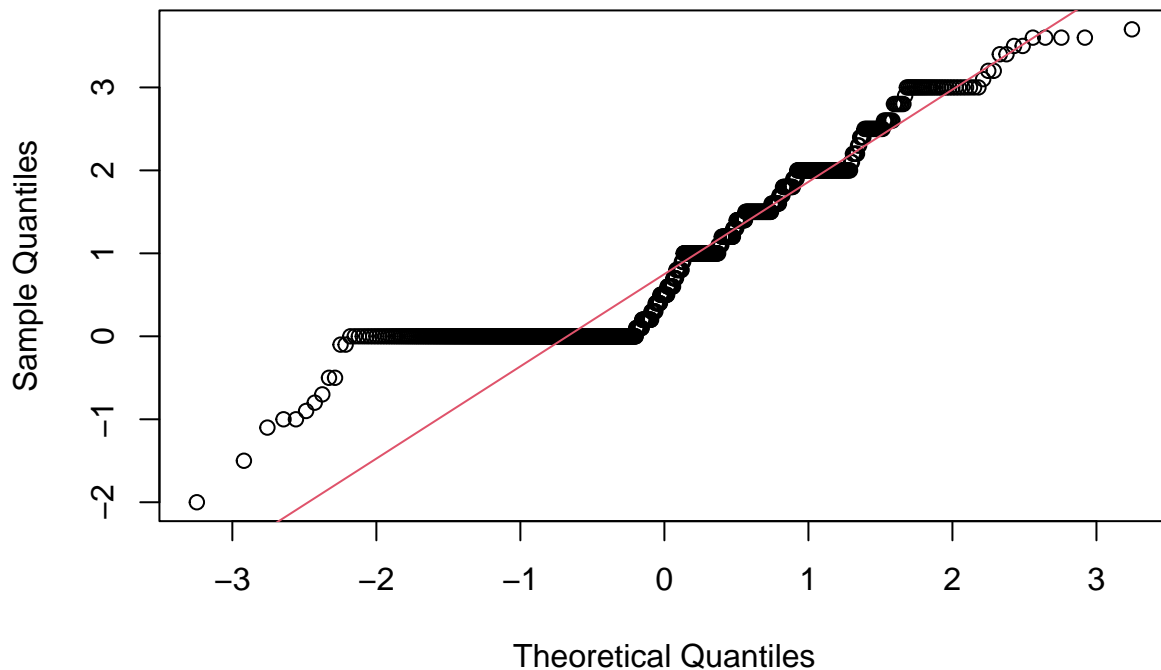
Q-Q para la variable: COLESTEROL



Q-Q para la variable: FREQ CARDÍACA MÁX



Q-Q para la variable: OLDPEAK



Vemos que tanto la distribución de los valores de las características de EDAD y de la Frecuencia Cardíaca máxima se acercan mucho a la normalidad, por otro lado la distribución de los valores de la característica Presión Arterial, Colesterol y Old distan de la normal.

Pruebas estadísticas

¿Que variables cuantitativas ejercen mayor influencia en la variable que define si hay una enfermedad cardíaca

```
corr_matrix <- matrix(nc=2, nr=0)
colnames(corr_matrix) <- c("estimate", "p-value")

# Calculamos el coeficiente de correlacion para cada variable cuantitativa
# con respecto al campo E. CARDICA

for(i in 1:(ncol(datos_final) -1 ))
{
  if(vector_tipos[i] == "numeric")
  {
    spearman_test = cor.test(unlist(datos_final[,i]),
                             unlist(datos_final[,length(datos_final)]),
                             method = "spearman")
    corr_coef <- spearman_test$estimate
```

```

    p_val <- spearman_test$p.value

    # Añade a la matriz
    pair <- matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(datos_final)[i]
  }
}

print(corr_matrix)

```

```

##              estimate      p-value
## EDAD          0.3013799 1.965724e-19
## PRESIÓN ARTERIAL 0.1013138 3.002400e-03
## COLESTEROL      -0.0549291 1.082846e-01
## NIVEL DE AZÚCAR  0.2784974 1.032387e-16
## FREC CARDÍACA MÁX -0.4156083 4.470311e-37
## OLDPEAK         0.4216309 3.238849e-38

```

Podemos identificar cual es la variable más correlacionada con la variable Enfermedad Cardíaca, viendo cuales de los valores de la columna estimate se acercan más al valor +1 o -1, en este caso los más cercanos y por lo tanto los que más correlacionados están con la variable objetivo son: OLDPEAK y FREC CARDÍACA MÁX.

Por otro lado en la columna p-value, tenemos el indicador del peso estadístico de cada variable, en este caso las variables que tienen peso estadístico más alto son: COLESTEROL y PRESIÓN ARTERIAL.

Grupos de datos

Podemos establecer una serie de grupos partiendo de nuestro dataset para analizar y/o compararlos más adelante

```

# Agrupación por Tipo de dolor de torax
# [0: TA; 1: ATA ; 2: NAP ; 3: ASY]
datos_final.angina_tipica <- datos_final[datos_final$`TIPO DOLOR TORAX` == 0,]
datos_final.angina_atipica <- datos_final[datos_final$`TIPO DOLOR TORAX` == 1,]
datos_final.no_angina <- datos_final[datos_final$`TIPO DOLOR TORAX` == 2,]
datos_final.asintomatico <- datos_final[datos_final$`TIPO DOLOR TORAX` == 3,]

# Agrupación por pacientes con Hiperglucemia
datos_final.sin_hiperglucemia <- datos_final[datos_final$`NIVEL DE AZÚCAR` == 0,]
datos_final.con_hiperglucemia <- datos_final[datos_final$`NIVEL DE AZÚCAR` == 1,]

# Agrupación por pacientes con Angina inducida
datos_final.sin_angina_inducida <- datos_final[datos_final$`ANGINA x EJERCICIO` == 0,]
datos_final.con_angina_inducida <- datos_final[datos_final$`ANGINA x EJERCICIO` == 1,]

# Agrupación por ECG en reposo
# ["Normal": 0, "ST": 1, "LVH": 2]

```



```
datos_final.ecg_normal <- datos_final[datos_final$`ECG EN REPOSO` == 0,]
datos_final.ecg_st <- datos_final[datos_final$`ECG EN REPOSO` == 1,]
datos_final.ecg_lvh <- datos_final[datos_final$`ECG EN REPOSO` == 2,]
```

¿La Presión arterial y su influencia en la enfermedad cardiaca?

Vamos a realizar una prueba estadística para establecer un contraste de hipótesis sobre dos muestras (una con presión arterial alta y otra con presión arterial normal) y ver cual de ellas tiene mayor probabilidades de sufrir enfermedad cardiaca.

Modelo Arbol de Decision

Podemos elaborar un arbol de decisión, para ver que variables tienen más influencia en la enfermedad cardiaca y establecer las reglas que definen dicha variable.

Test estadísticos de significancia

Antes de proceder a la clasificación de los parámetros de pacientes con más probabilidades de sufrir enfermedad cardiaca, deberemos de hacer una selección previa de las características a utilizar en nuestro modelo.

Para ello nos vamos a ayudar de una matriz de correlación con el objetivo de confirmar las conclusiones en cuanto a correlación de variables obtenidas en el apartado anterior.

```
if(!require("DescTools"))
{
  install.packages("DescTools")
  # http://cran.us.r-project.org
  library("DescTools")
}
```

```
## Loading required package: DescTools
```

```
## Warning: package 'DescTools' was built under R version 4.2.2
```

```
# Analizamos las correlaciones de todas las características de tipo categoricas con "E. CARDIACA"
# Lo añadimos a una tabla

datos_corr.Phi <- list()
datos_corr.CramerV <- list()
datos_corr.nombre <- list()

ind <- 1
for (i in colnames(datos_final))
{
  if(i != "E. CARDIACA")
  {
    tabla_cruzada <- table(as.numeric(unlist(datos_final[i])), datos_final$`E. CARDIACA`)
    datos_corr.CramerV <- append(datos_corr.CramerV, CramerV(tabla_cruzada))
```

```

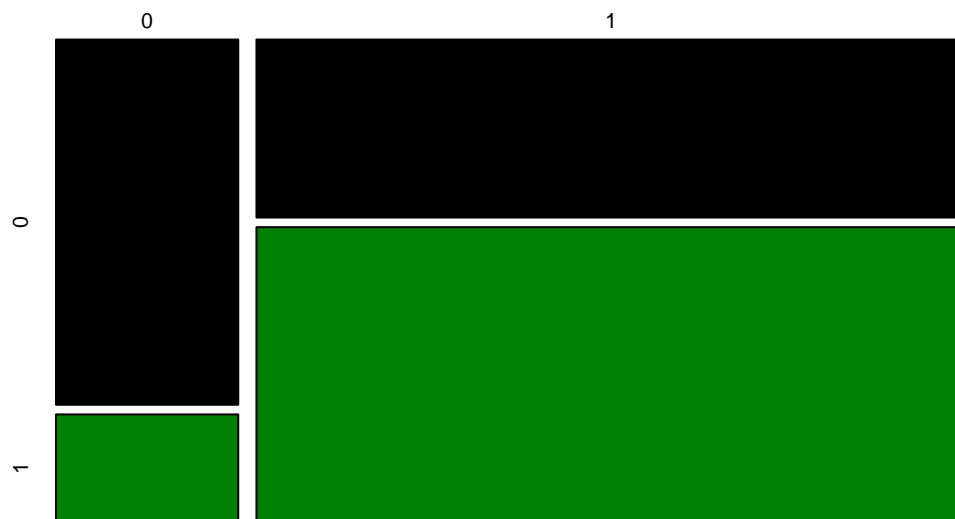
datos_corr.Phi <- append(datos_corr.Phi, Phi(tabla_cruzada))
datos_corr.nombre <- append(datos_corr.nombre, i)
}

if (vector_tipos[ind] != 'numeric')
{
  # Solo pintamos las variables categoricas ya que con las de tipo numerico no se aprecian los valores
  plot(tabla_cruzada, col = c("black", "#008000"), main = paste0(i, " vs. E. CARDIACA"))
}

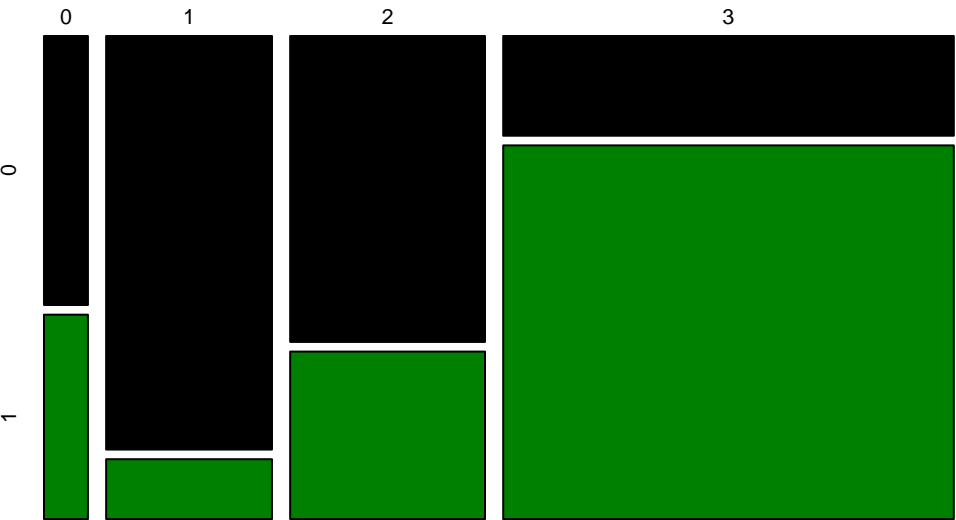
ind <- ind + 1
}

```

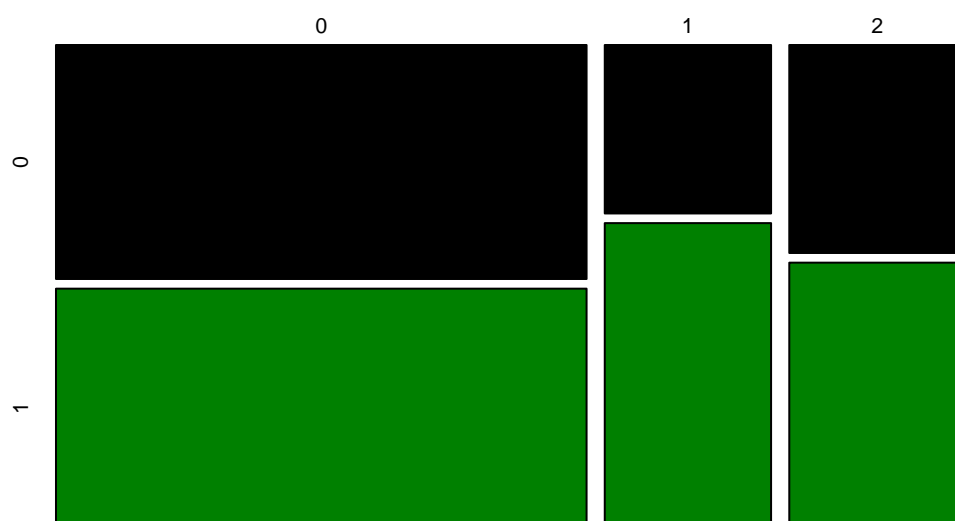
SEXO vs. E. CARDIACA



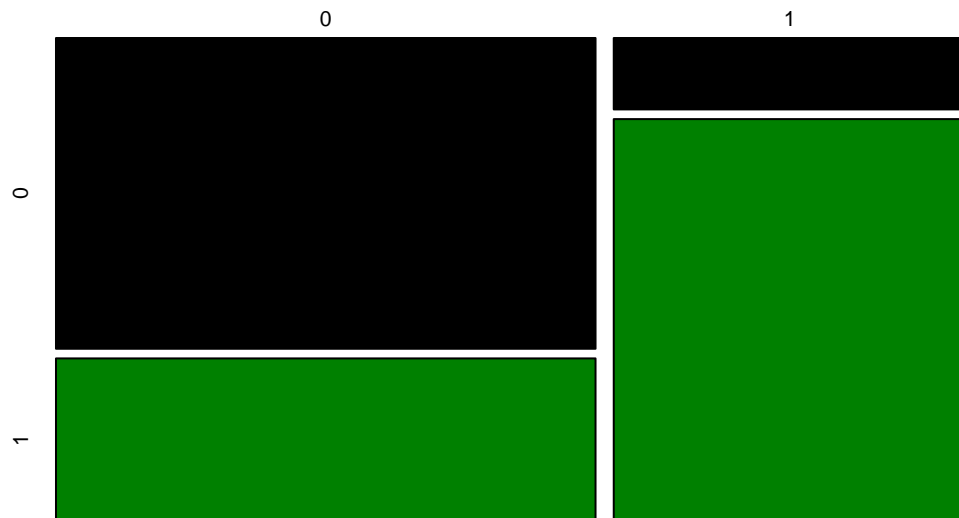
TIPO DOLOR TORAX vs. E. CARDIACA



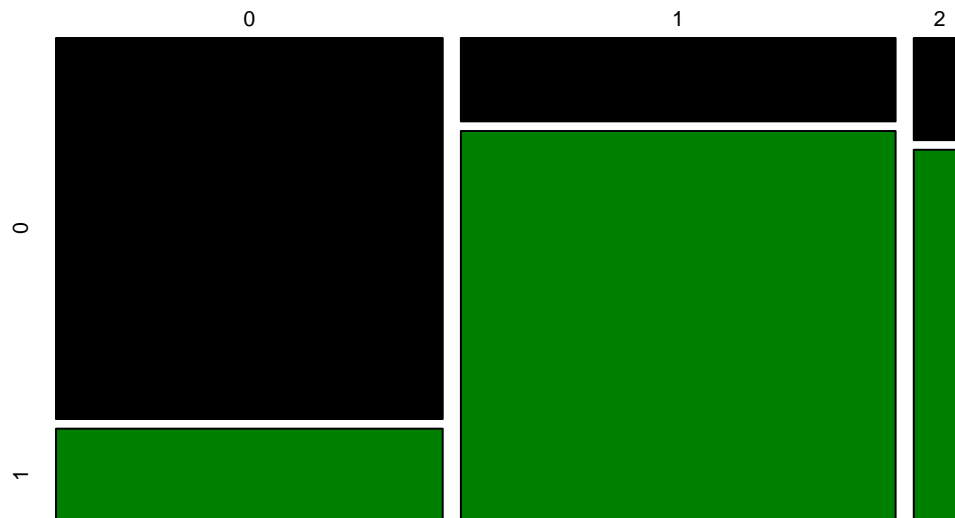
ECG EN REPOSO vs. E. CARDIACA



ANGINA x EJERCICIO vs. E. CARDIACA



PENDIENTE ST vs. E. CARDIACA



```
n_list <- list(nombre=as.character(datos_corr.nombre),
               CamerV=as.numeric(datos_corr.CamerV),
               Phi=as.numeric(datos_corr.Phi))

df_CramerV <- (as.data.frame(do.call(cbind, n_list)))
print(df_CramerV[order(df_CramerV$Phi, decreasing = TRUE),])
```

##	nombre	CamerV	Phi
## 11	PENDIENTE ST	0.622951452964884	0.622951452964884
## 5	COLESTEROL	0.591523072486335	0.591523072486335
## 3	TIPO DOLOR TORAX	0.547585583133845	0.547585583133845
## 8	FREC CARDÍACA MÁX	0.527523642518057	0.527523642518057
## 10	OLDPEAK	0.501978806690108	0.501978806690108
## 9	ANGINA x EJERCICIO	0.496038059644464	0.496038059644464
## 1	EDAD	0.383589901936776	0.383589901936776
## 2	SEXO	0.320210873993129	0.320210873993129
## 4	PRESIÓN ARTERIAL	0.314708115437243	0.314708115437243
## 6	NIVEL DE AZÚCAR	0.278497410608853	0.278497410608853
## 7	ECG EN REPOSO	0.107253870350395	0.107253870350395

Valores de la V de Cramér (https://en.wikipedia.org/wiki/Cramér%27s_V) y Phi (https://en.wikipedia.org/wiki/Phi_coefficient) entre 0.1 y 0.3 nos indican que la asociación estadística es baja, y entre 0.3 y 0.5 se puede considerar una asociación media. Finalmente, si los valores fueran superiores a 0.5 la asociación estadística entre las variables sería alta.

Podemos observar dentro del dataframe: df_cramerV que las variables PENDIENTES ST, COLESTEROL, TIPO DOLOR TORAX, FREC CARDIACA MAX y OLDPEAK tienen correlación alta con E. CARDIACA.

Usaremos dichas variables para la obtención del Arbol de decisión, se podría utilizar un número mayor de variables, pero podría hacerse mucho complejo (con muchas reglas de decisión)

```
# características con significancia estadística:
nombres_columnas <- df_CramerV$nombre[df_CramerV$Phi > 0.5]
nombres_columnas

## [1] "TIPO DOLOR TORAX" "COLESTEROL" "FREC CARDÍACA MÁX"
## [4] "OLDPEAK" "PENDIENTE ST"
```

Aplicación del modelo Decision Tree (Arbol de decisión)

Aplicamos el modelo Decision Tree sobre las 3 características que hemos obtenido en el test de significancia estadística de Cramer V.

```
# Backup dataset inicial:
datos_final_orig <- datos_final

# Convertimos todas las variables a tipo factor
datos_final[] <- lapply(datos_final, factor)

# Reducimos el dataset
for(i in colnames(datos_final))
{
  if(!i %in% nombres_columnas)
  {
    if (!i == "E. CARDIACA")
    {
      datos_final$i <- NULL
    }
  }
}
set.seed(666)
y <- datos_final$`E. CARDIACA` # Variable objetivo
X <- datos_final[nombres_columnas] # Variables predictoras
```

Separamos conjunto de test y de entrenamiento con una proporción 33% Test 66% Training.

```
split_prop <- 3
indexes = sample(1:nrow(X), size=floor(((split_prop-1)/split_prop)*nrow(X)))
train_X <- X[indexes,]
train_y <- y[indexes]
test_X <- X[-indexes,]
test_y <- y[-indexes]
```

Comprobamos que las variables train_y y test_y estén balanceadas reflejo de la variable y.

```
print("y:")
```

```
## [1] "y:"
```

```
summary(y)
```

```
##    0    1  
## 391 465
```

```
print("train_y:")
```

```
## [1] "train_y:"
```

```
summary(train_y)
```

```
##    0    1  
## 264 306
```

```
print("test_y:")
```

```
## [1] "test_y:"
```

```
summary(test_y)
```

```
##    0    1  
## 127 159
```

Hacemos lo mismo con las variables train_crX y test_crX y X

```
print("X:")
```

```
## [1] "X:"
```

```
summary(X)
```

```
## TIPO DOLOR TORAX    COLESTEROL  FREQ CARDÍACA MÁX    OLDPEAK    PENDIENTE ST  
## 0: 44              198      :167   150      : 38        0          :349   0:379  
## 1:166              254      : 11   140      : 35        1          : 80   1:426  
## 2:195              220      : 10   120      : 32        2          : 69   2: 51  
## 3:451              223      : 10   130      : 28        1.5        : 50  
##                   204      :  9   160      : 24        3          : 28  
##                   211      :  9   110      : 22        1.2        : 25  
##                   (Other):640  (Other):677  (Other):255
```

```
print("train_X:")
```

```
## [1] "train_X:"
```



```
summary(train_X)
```

```
## TIPO DOLOR TORAX COLESTEROL FREC CARDÍACA MÁX OLDPEAK PENDIENTE ST
## 0: 33 198 :110 140 : 26 0 :239 0:251
## 1:110 254 : 8 130 : 21 1 : 54 1:285
## 2:139 223 : 7 150 : 21 2 : 41 2: 34
## 3:288 240 : 7 160 : 18 1.5 : 33
## 204 : 6 170 : 16 1.2 : 16
## 207 : 6 110 : 14 3 : 16
## (Other):426 (Other):454 (Other):171
```

```
print("test_X:")
```

```
## [1] "test_X:"
```

```
summary(test_X)
```

```
## TIPO DOLOR TORAX COLESTEROL FREC CARDÍACA MÁX OLDPEAK PENDIENTE ST
## 0: 11 198 : 57 120 : 18 0 :110 0:128
## 1: 56 220 : 7 150 : 17 2 : 28 1:141
## 2: 56 230 : 5 140 : 9 1 : 26 2: 17
## 3:163 234 : 5 110 : 8 1.5 : 17
## 195 : 4 115 : 7 3 : 12
## 211 : 4 122 : 7 1.2 : 9
## (Other):204 (Other):220 (Other): 84
```

Se crea el árbol de decisión usando los datos de entrenamiento (no hay que olvidar que la variable outcome es de tipo factor):

```
tree <- C50::C5.0(train_X, train_y, rules=TRUE )
summary(tree)
```

```
##
## Call:
## C5.0.default(x = train_X, y = train_y, rules = TRUE)
##
## C5.0 [Release 2.07 GPL Edition] Wed Jan 4 12:21:55 2023
## -----
##
## Class specified by attribute 'outcome'
##
## Read 570 cases (6 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (251/47, lift 1.7)
## PENDIENTE ST = 0
## -> class 0 [0.810]
##
```

```

## Rule 2: (319/60, lift 1.5)
## PENDIENTE ST in {1, 2}
## -> class 1 [0.810]
##
## Default class: 1
##
##
## Evaluation on training data (570 cases):
##
##      Rules
##      -----
##      No      Errors
##
##      2  107(18.8%)  <<
##
##      (a)  (b)  <-classified as
##      ----  ----
##      204   60   (a): class 0
##      47   259   (b): class 1
##
##
## Attribute usage:
##
## 100.00% PENDIENTE ST
##
##
## Time: 0.0 secs

```

El modelo decision tree explica con dos reglas la probabilidad de sufrir una enfermedad cardiaca en función de las variables: TIPO DOLOR TORAX, COLESTEROL FREC CARDÍACA MÁX, OLDPEAK, PENDIENTE ST.

Regla: 1 -> PENDIENTE ST = 0 -> Tendrán menos probabilidades de sufrir E. CARDIACA Regla: 2 -> PENDIENTE ST in {1, 2} -> Tendrán más probabilidades de sufrir E. CARDIACA

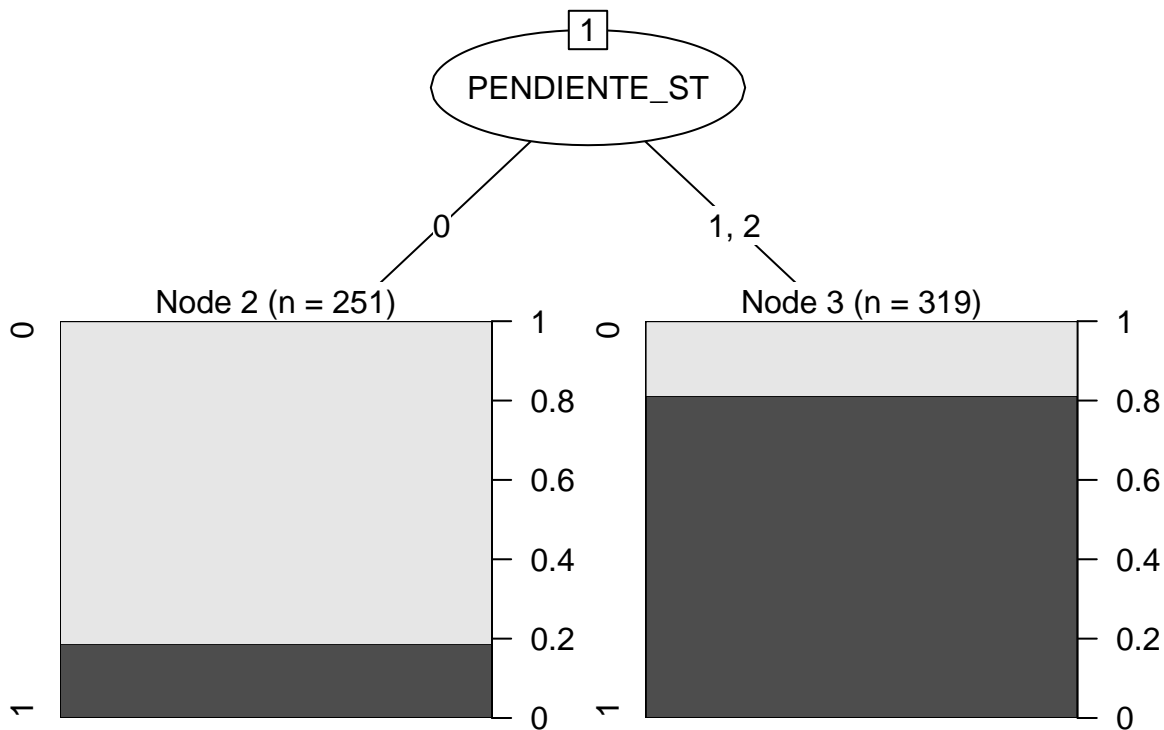
El modelo solo usa la variable predictora PENDIENTE ST, y tiene una tasa de error de 18.4 % es decir es capaz de explicar el 82.6 % de los casos.

De manera más gráfica:

```

# RENOMBAMOS VARIABLES PARA PODER DIBUJAR:
train_X2 <- train_X
colnames(train_X2)[1] <- "TIPO_DOLOR_TORAX"
colnames(train_X2)[3] <- "FREC_CARD_MAX"
colnames(train_X2)[5] <- "PENDIENTE_ST"
model <- C50::C5.0(train_X2, train_y)
plot(model)

```



Como podemos observar de manera visual el modelo basado en árbol de decisión, solo tiene en cuenta la variable “PENDIENTE_ST”, para decidir entre si un paciente es propenso a sufrir una enfermedad cardíaca o no.

Evaluación del modelo árbol de decision

Una vez tenemos el modelo, podemos comprobar su calidad prediciendo la clase para los datos de prueba que nos hemos reservado al principio.

```
predicted_model <- predict( tree, test_X, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%", 100*sum(predicted_model == test_y) / length(predicted_model)))
```

```
## [1] "La precisión del árbol es: 81.4685 %"
```

Cuando hay pocas clases, la calidad de la predicción se puede analizar mediante una matriz de confusión que identifica los tipos de errores cometidos.

```
mat_conf <- table(test_y, Predicted=predicted_model)
mat_conf
```

```
##      Predicted
## test_y    0    1
##      0 101  26
##      1  27 132
```

De la matriz de confusión observamos los siguientes valores:

Verdaderos Negativos (E. CARDIACA): 110 Verdaderos Positivos (E. CARDIACA): 123 Falsos Negativos (E. CARDIACA): 28 Falsos Positivos (E. CARDIACA): 25

El modelo podría mejorarse sesgando a minimizar los falsos negativos, ya que no queremos que se nos escapen del diagnóstico pacientes que puedan desarrollar una enfermedad cardíaca.

Random Forest

Nos interesa saber para las predicciones qué variable son las que tienen más influencia. Así, probaremos con un enfoque algorítmico de Random Forest y obtendremos métricas de interpretabilidad con la librería IML (<https://cran.r-project.org/web/packages/iml/iml.pdf>). Así:

```
if(!require(randomForest)){  
  install.packages('randomForest', repos='http://cran.us.r-project.org')  
  library(randomForest)  
}
```

```
## Loading required package: randomForest  
  
## Warning: package 'randomForest' was built under R version 4.2.2  
  
## randomForest 4.7-1.1  
  
## Type rfNews() to see new features/changes/bug fixes.  
  
##  
## Attaching package: 'randomForest'  
  
## The following object is masked from 'package:dplyr':  
##  
##      combine  
  
## The following object is masked from 'package:ggplot2':  
##  
##      margin
```

```
if(!require(iml)){  
  install.packages('iml', repos='http://cran.us.r-project.org')  
  library(iml)  
}
```

```
## Loading required package: iml  
  
## Warning: package 'iml' was built under R version 4.2.2
```

Discretización de variables:

```
# Antes de aplicar el modelo random forest con el conjunto de variables obtenidos para el arbol de decisi
# Establremos los siguientes niveles:
# COLESTEROL [ COL: MENOR DE 200 -> 0, COL: ENTRE 200 - 400 -> 1, COL: ENTRE 400 - 600 -> 2, COL: MAYOR
```

```
colnames(train_X2)
```

```
## [1] "TIPO_DOLOR_TORAX" "COLESTEROL"          "FREC_CARD_MAX"      "OLDPEAK"
## [5] "PENDIENTE_ST"
```

```
train.data <- as.data.frame(cbind(train_X2[c(1,5,4)] ,train_y))
colnames(train.data)[4] <- "ENF_CARDIACA"
rf <- randomForest(ENF_CARDIACA ~ ., data = train.data, ntree = 50)

X <- train.data[which(names(train.data) != "ENF_CARDIACA")]
predictor <- Predictor$new(rf, data = X, y = train.data$ENF_CARDIACA)
imp <- FeatureImp$new(predictor, loss = "ce")
```

```
## Warning: package 'C50' was built under R version 4.2.2

## Warning: package 'rpart' was built under R version 4.2.2

## Warning: package 'tidyselect' was built under R version 4.2.2

## Warning: package 'expm' was built under R version 4.2.2

## Warning: package 'Cubist' was built under R version 4.2.2

## Warning: package 'checkmate' was built under R version 4.2.2

## Warning: package 'prediction' was built under R version 4.2.2

## Warning: package 'partykit' was built under R version 4.2.2

## Warning: package 'libcoin' was built under R version 4.2.2

## Warning: package 'lifecycle' was built under R version 4.2.2

## Warning: package 'Metrics' was built under R version 4.2.2

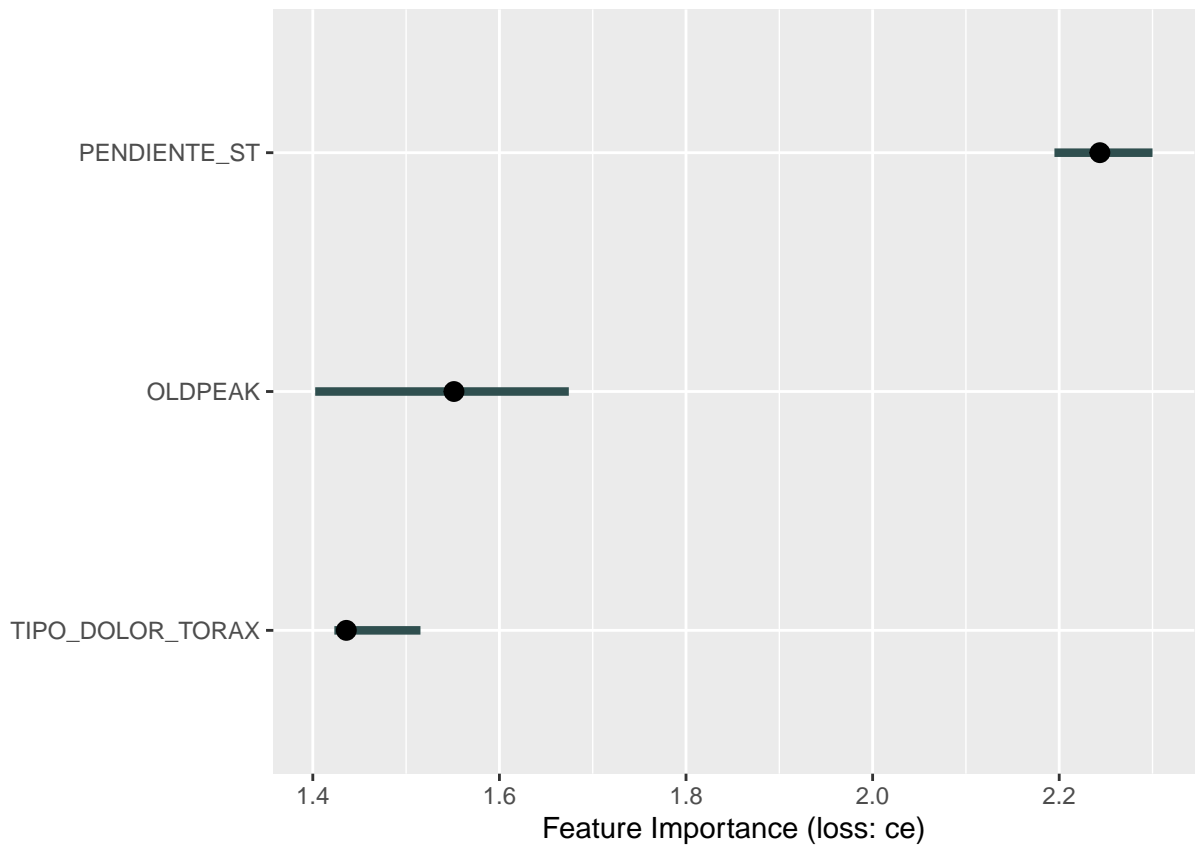
## Warning: package 'inum' was built under R version 4.2.2

## Warning: package 'igraph' was built under R version 4.2.2

## Warning: package 'gld' was built under R version 4.2.2

## Warning: package 'networkD3' was built under R version 4.2.2
```

```
plot(imp)
```

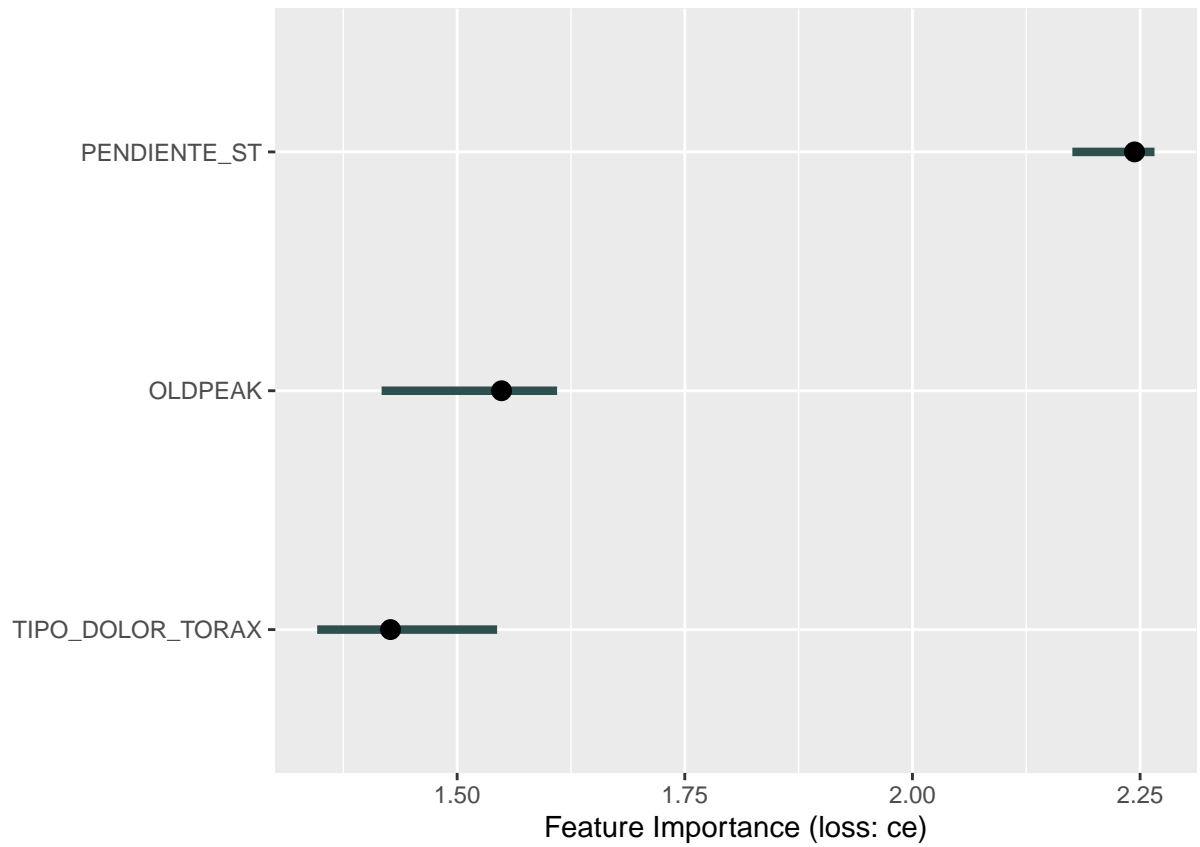


```
imp$results
```

```
##           feature importance.05 importance importance.95 permutation.error
## 1    PENDIENTE_ST      2.194872  2.243590      2.300000      0.3070175
## 2      OLDPEAK       1.402564  1.551282      1.674359      0.2122807
## 3 TIPO_DOLOR_TORAX    1.423077  1.435897      1.515385      0.1964912
```

Podemos medir y graficar la importancia de cada variable para las predicciones del random forest con *FeatureImp*. La medida se basa funciones de pérdida de rendimiento que en nuestro caso será con el objetivo de clasificación (“ce”).

```
X <- train.data[which(names(train.data) != "SURVIVED")]
predictor <- Predictor$new(rf, data = X, y = train.data$SURVIVED)
imp <- FeatureImp$new(predictor, loss = "ce")
plot(imp)
```



```
imp$results
```

```
##           feature importance.05 importance importance.95 permutation.error
## 1  PENDIENTE_ST      2.175610  2.243902    2.265854      0.3228070
## 2    OLDPEAK       1.417073  1.548780    1.609756      0.2228070
## 3 TIPO_DOLOR_TORAX  1.346341  1.426829    1.543902      0.2052632
```