¿Cómo realizar la limpieza y análisis de datos?

Autores: Eduardo Mora González y Diego Sánchez De La Fuente

Enero 2023

Contents

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 extraido 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 caracteristicas 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 anormalidad 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 datase
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 las 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                                         Mean   :132.4
##  3rd Qu.:60.00                                         3rd Qu.:140.0
##  Max.   :77.00                                         Max.   :200.0
##   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=="")
```

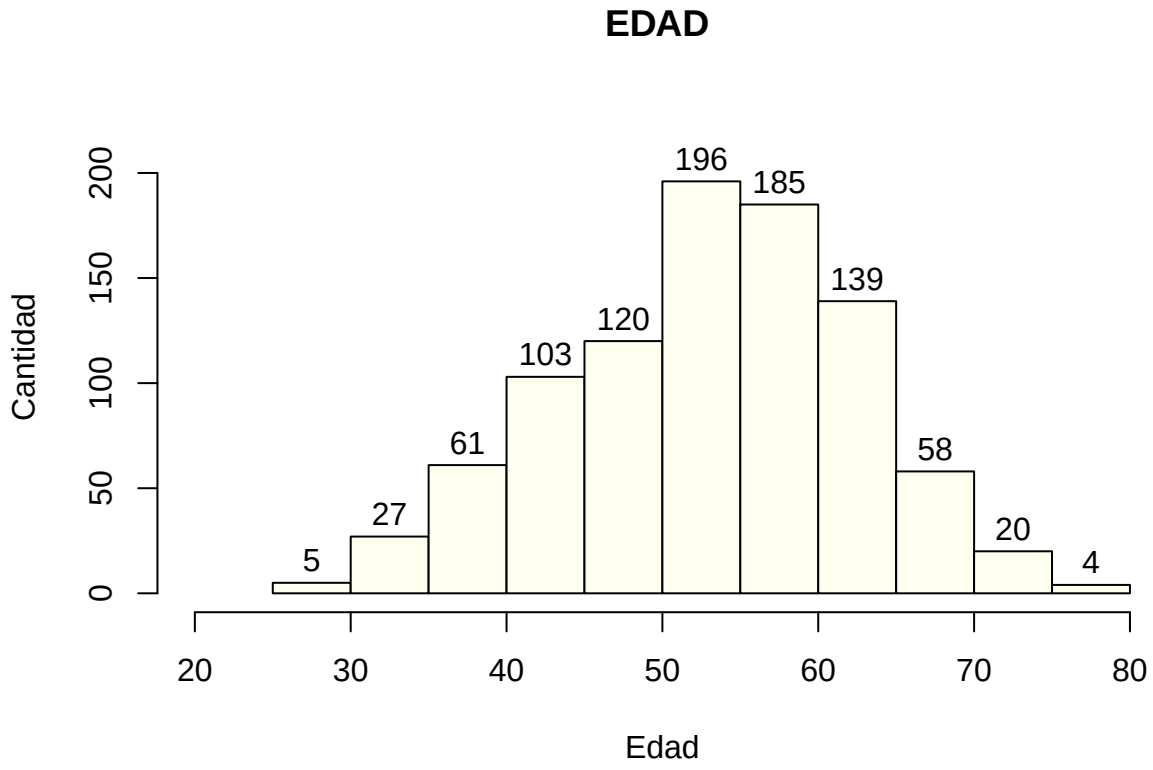
```
##             Age             Sex  ChestPainType      RestingBP    Cholesterol
##               0               0              0              0              0
##       FastingBS     RestingECG          MaxHR ExerciseAngina        Oldpeak
##               0               0              0        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

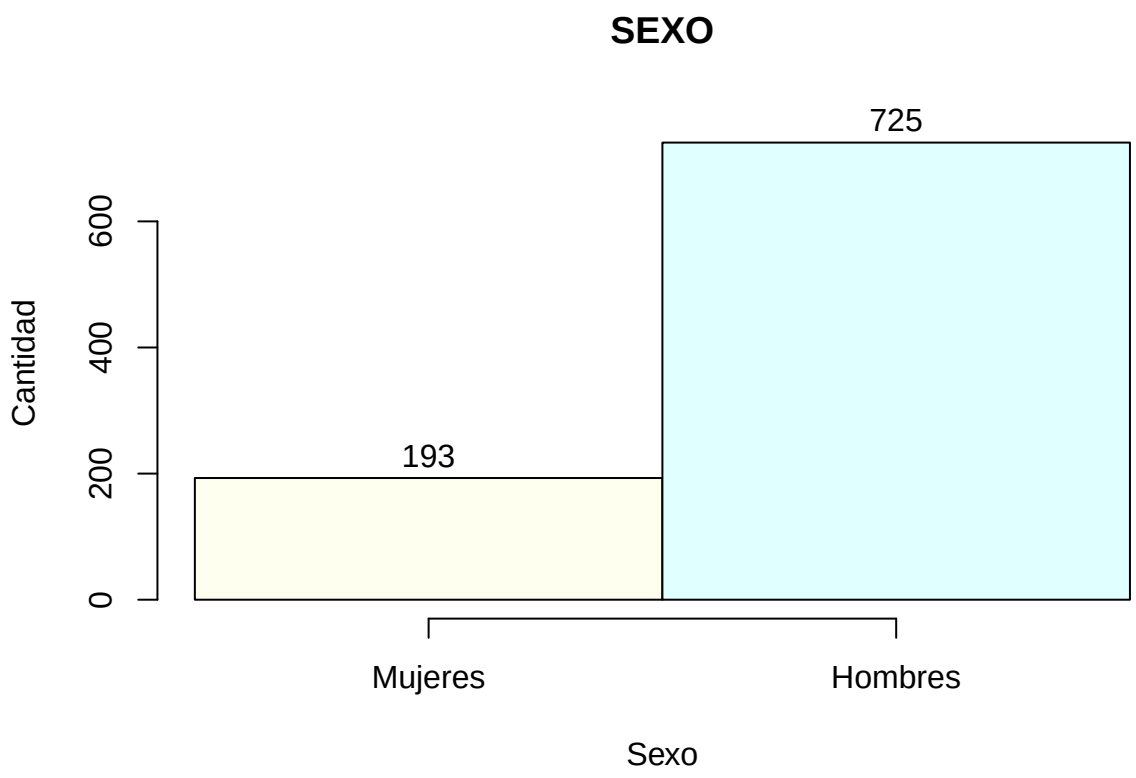
SEXO

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)
```



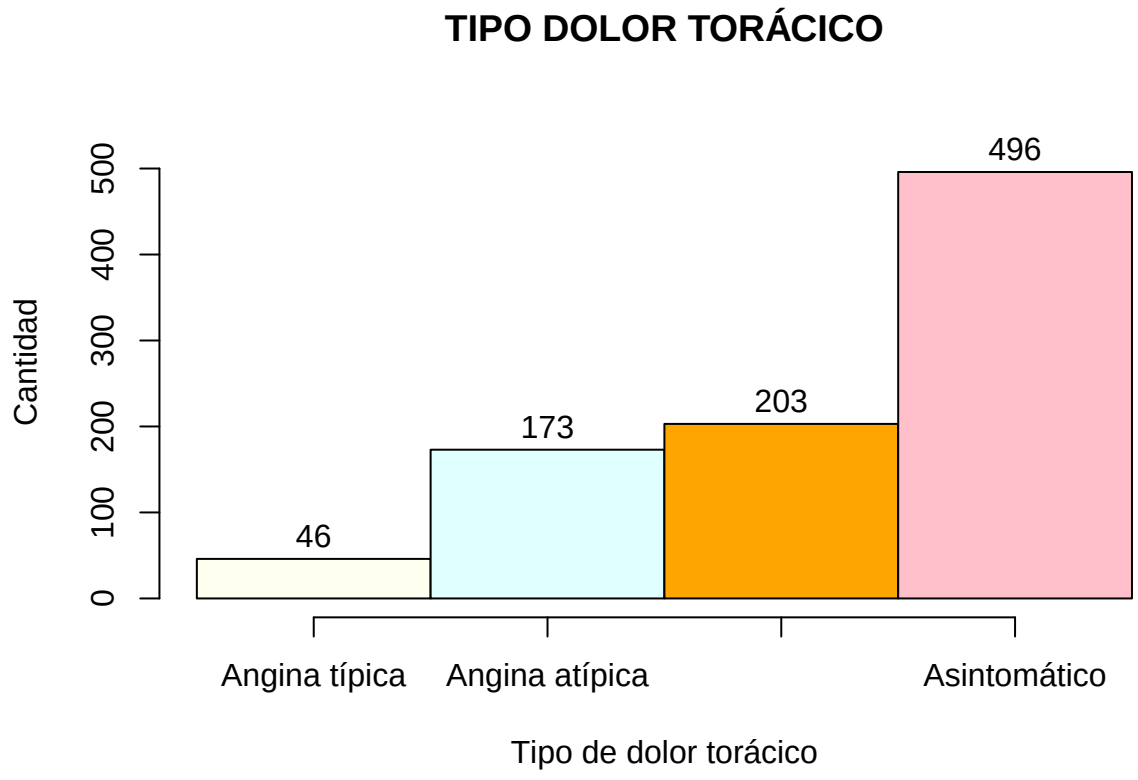
TIPO DE DOLOR TORÁCICO (ChestPainType)

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:

```r
#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.

```r
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
```

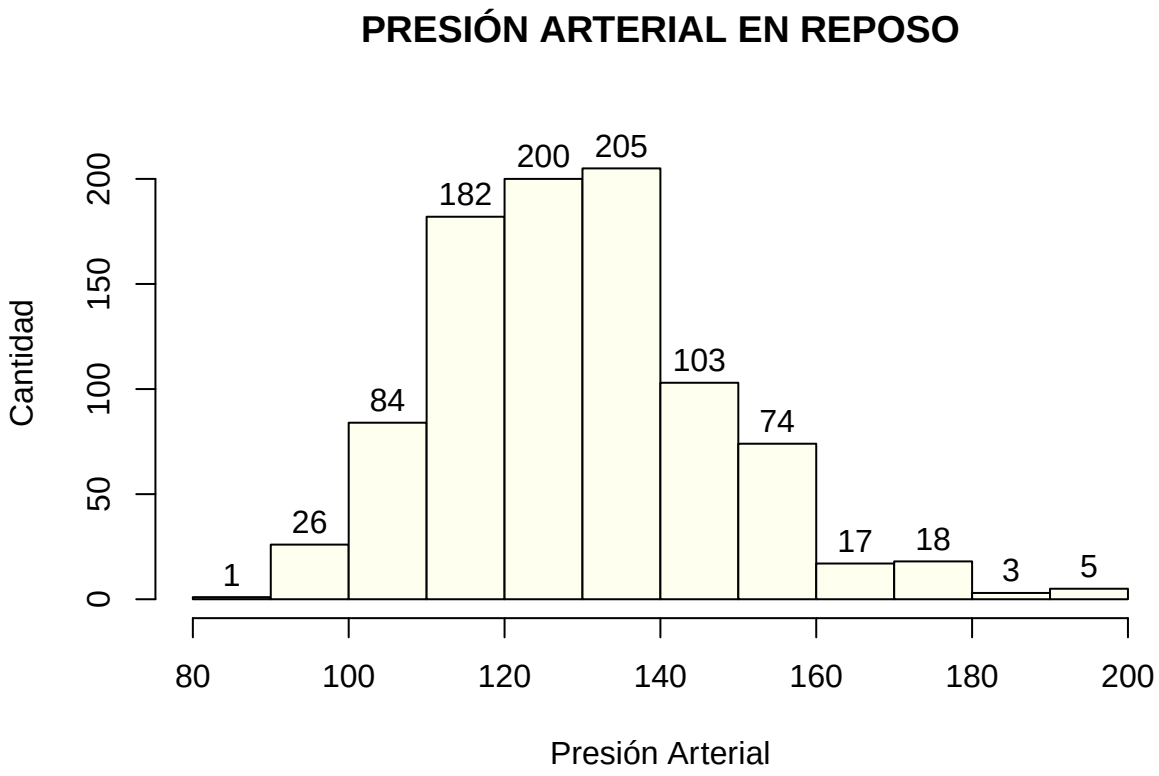
Como solo aparece una vez, se le asignará un valor por defecto. El valor por defecto será el más común.

```
#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 estaditicas 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))
```



COLESTEROL (Cholesterol)

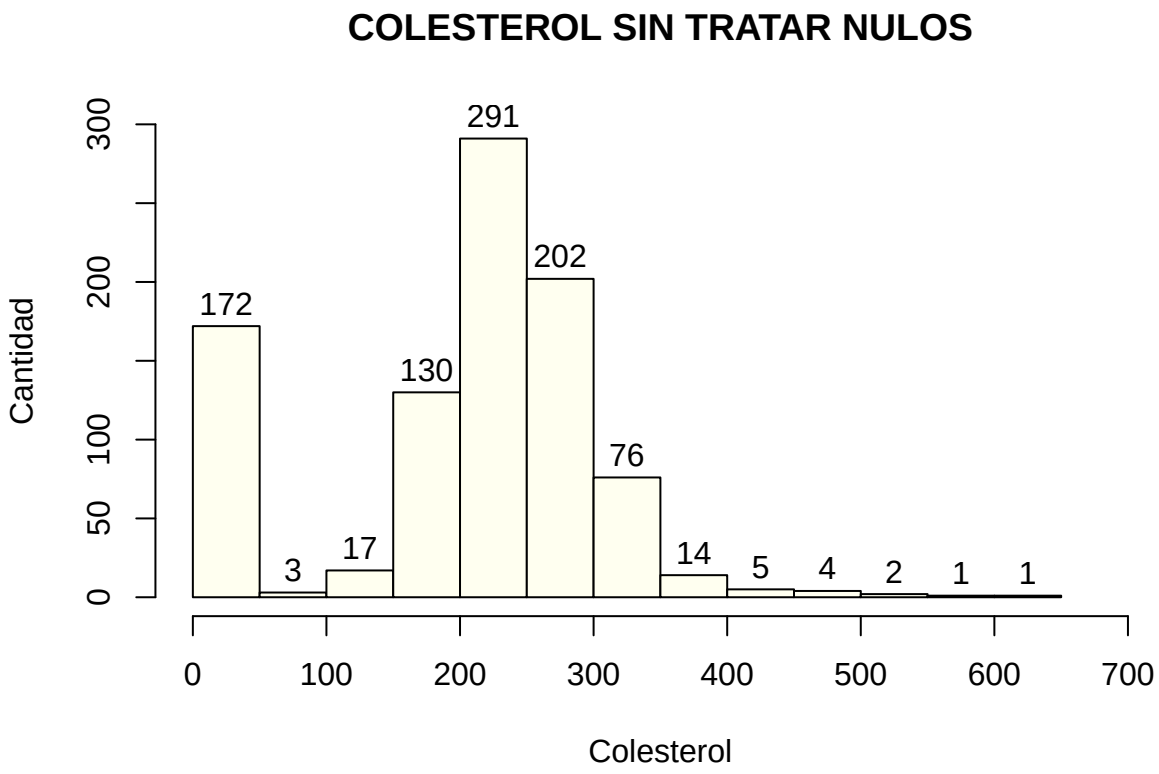
La siguiente característica es de tipo numérico. Al igual que en la presión arterial en reposo, que tenemos valores 0 que debemos analizar. Lo primero que se va a hacer es obtener el numero de casos que el colesterol es 0, y se consideraran las diversas formas de tratar estos datos.

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



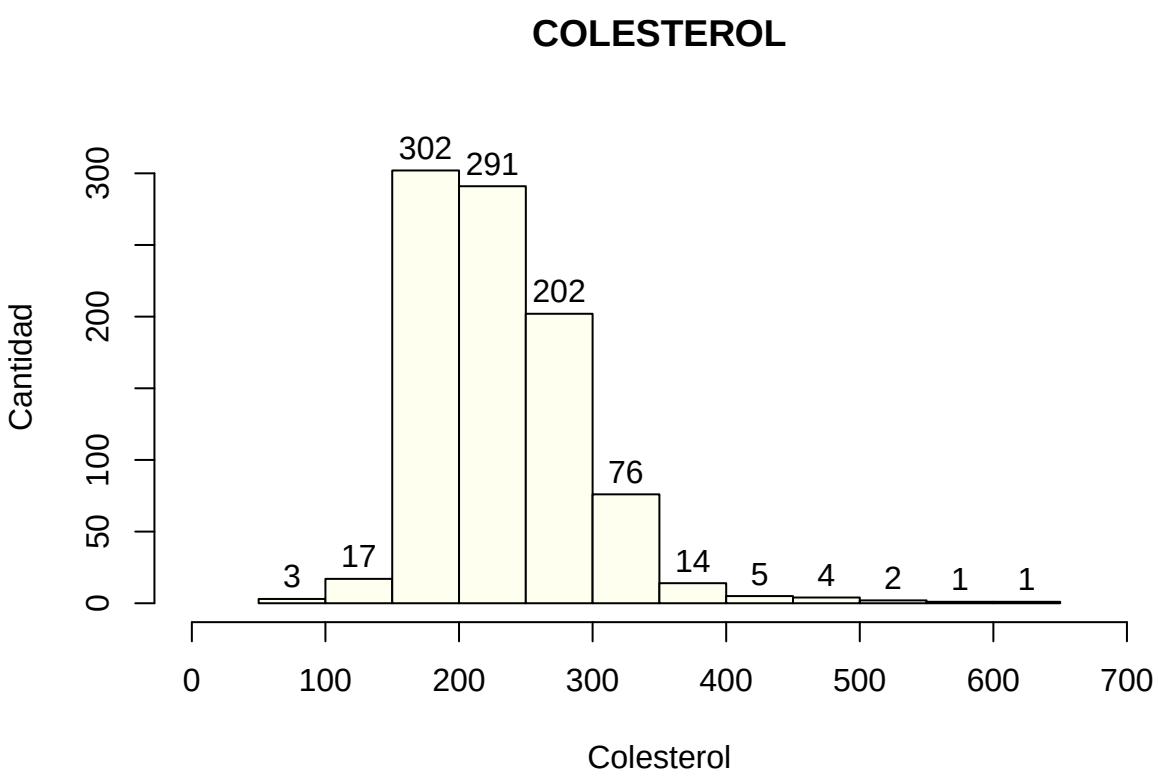
Tras analizar la gráfica y para no perder estos datos, se le asignaran un valor por defecto, que será la media de los datos. Esta decisión se ha tomado ya que poner el más común, nos crearía un conjunto de datos muy distintos entre unas medidas y otras, mientras que poner la media sería un valor que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

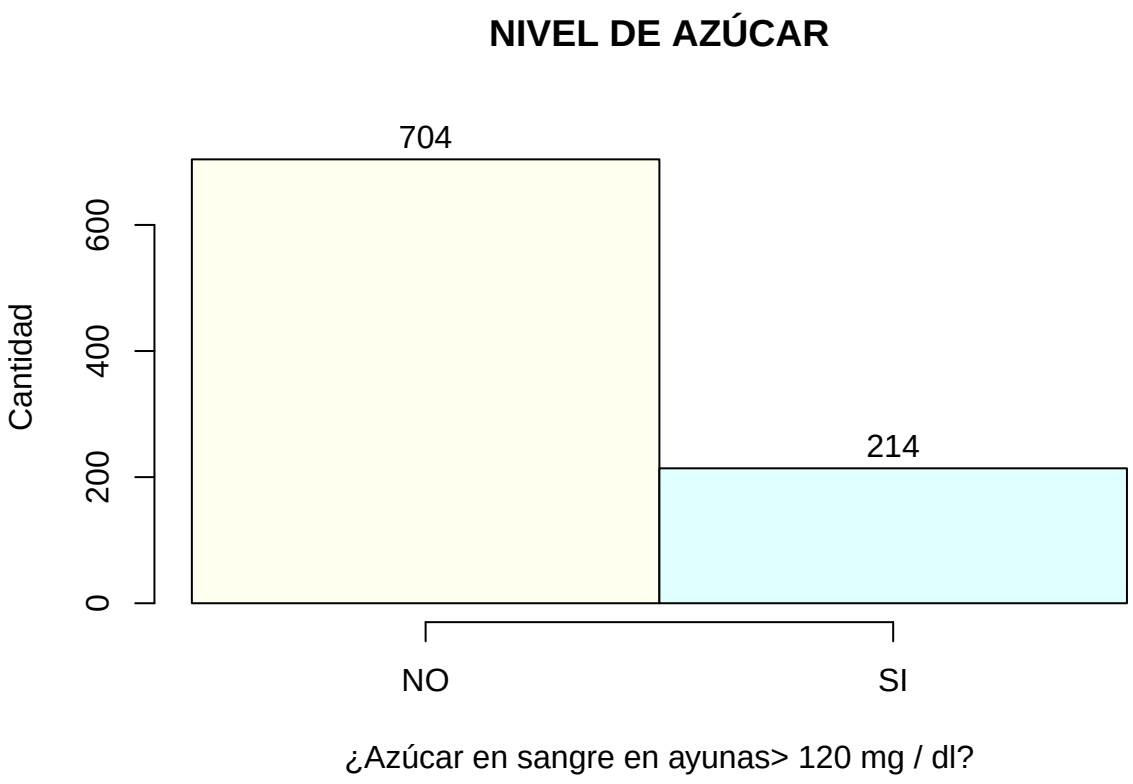


NIVEL DE AZÚCAR EN SANGRE EN AYUNAS (FastingBS)

Como se puede comprobar el conjunto de los datos puedes 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)

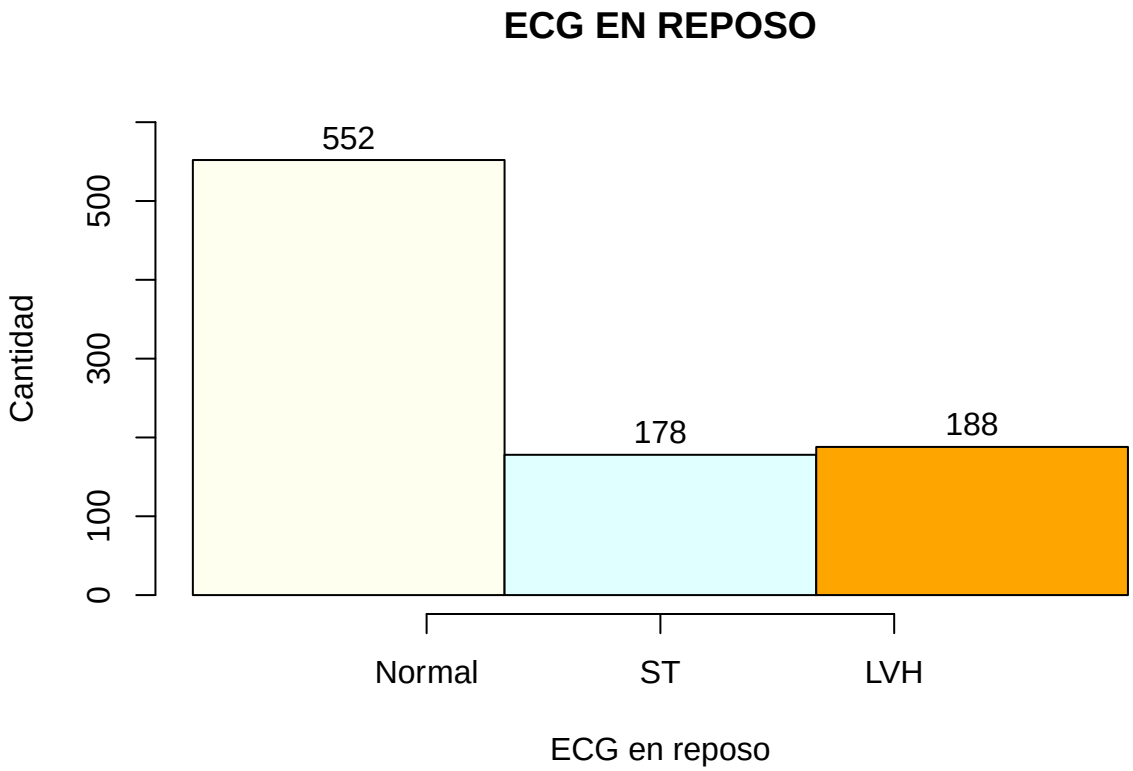
Nos damos cuenta de que el conjunto de datos viene identificado por 3 variables categóricas: + Normal: Normal, + ST: con anomalía de la onda ST-T + LVH: que muestra una hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes. Normalizamos para tenerlo de tipo numérico todas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

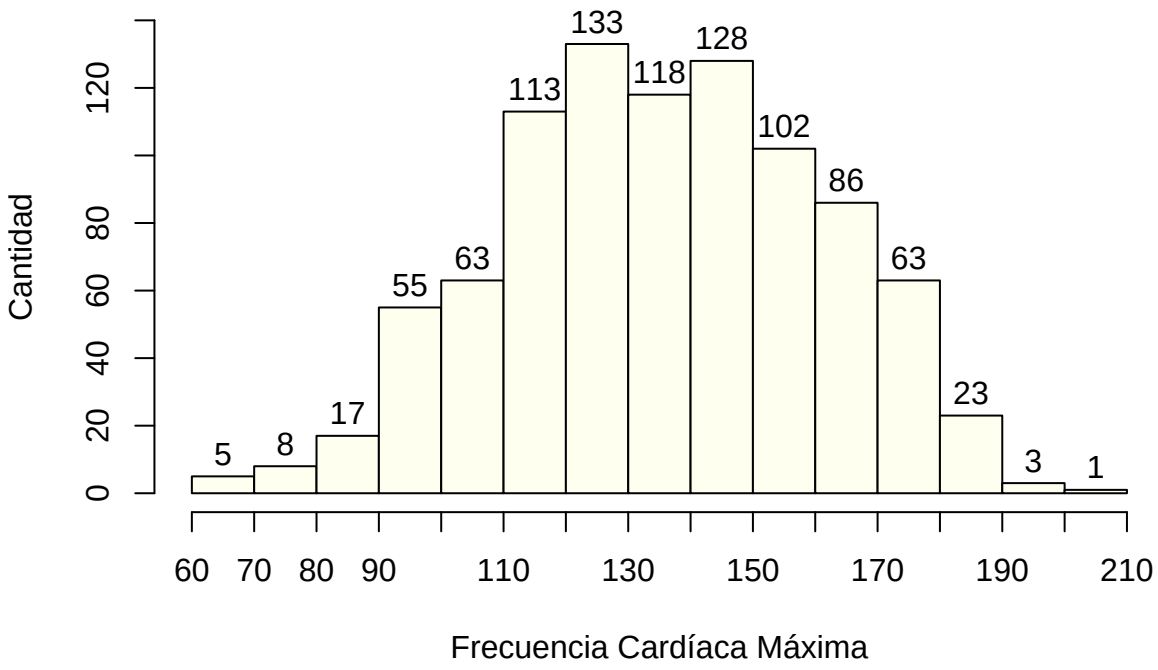
```
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 1, 1.75 ), cex.axis=1, labels = c("Normal","ST", "LVH"))
axis(2)
```



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)
```



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).

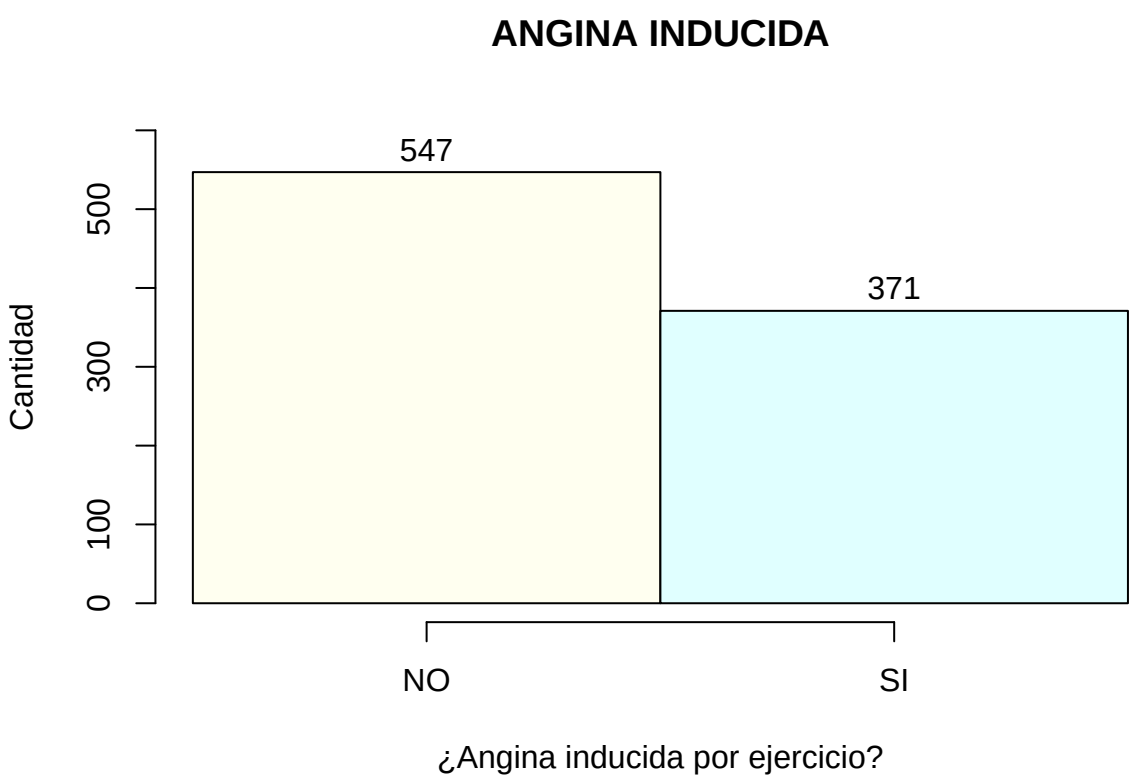
ANGINA INDUCIDA POR EJERCICIO (ExerciseAngina)

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)
```

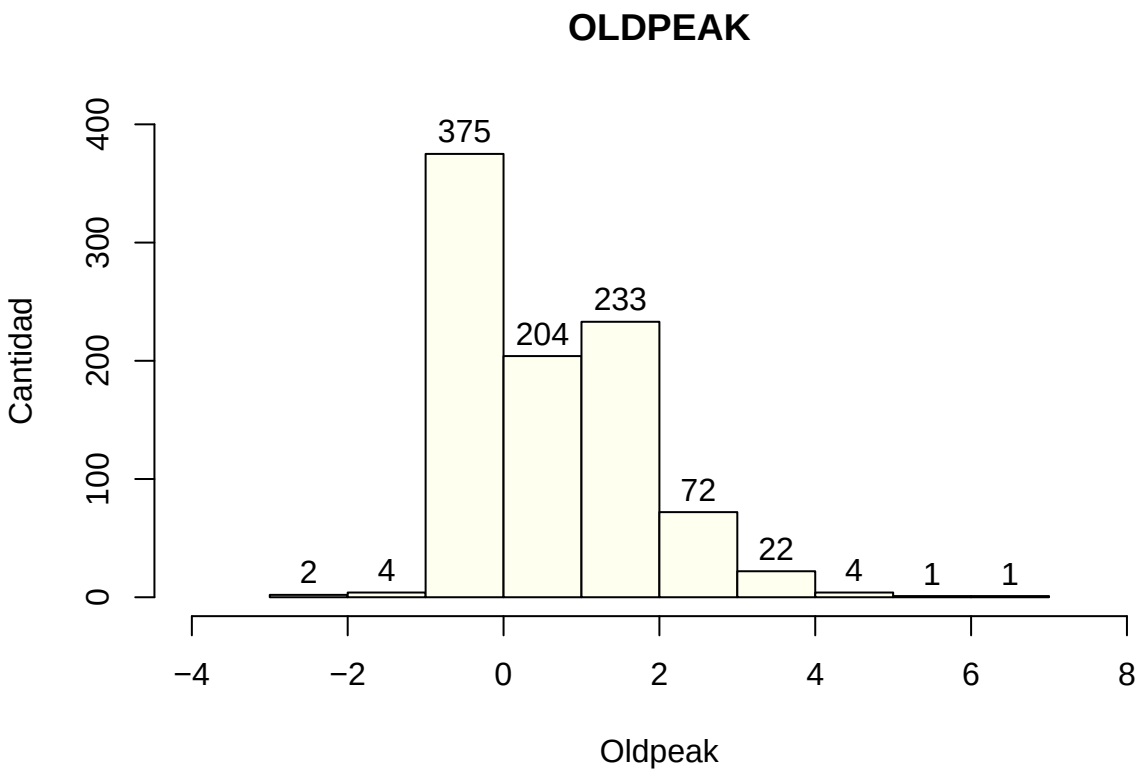


Como se puede apreciar, hay mas casos en que NO se ha producido una angina inducida por el ejercicio de que Si se haya producido.

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

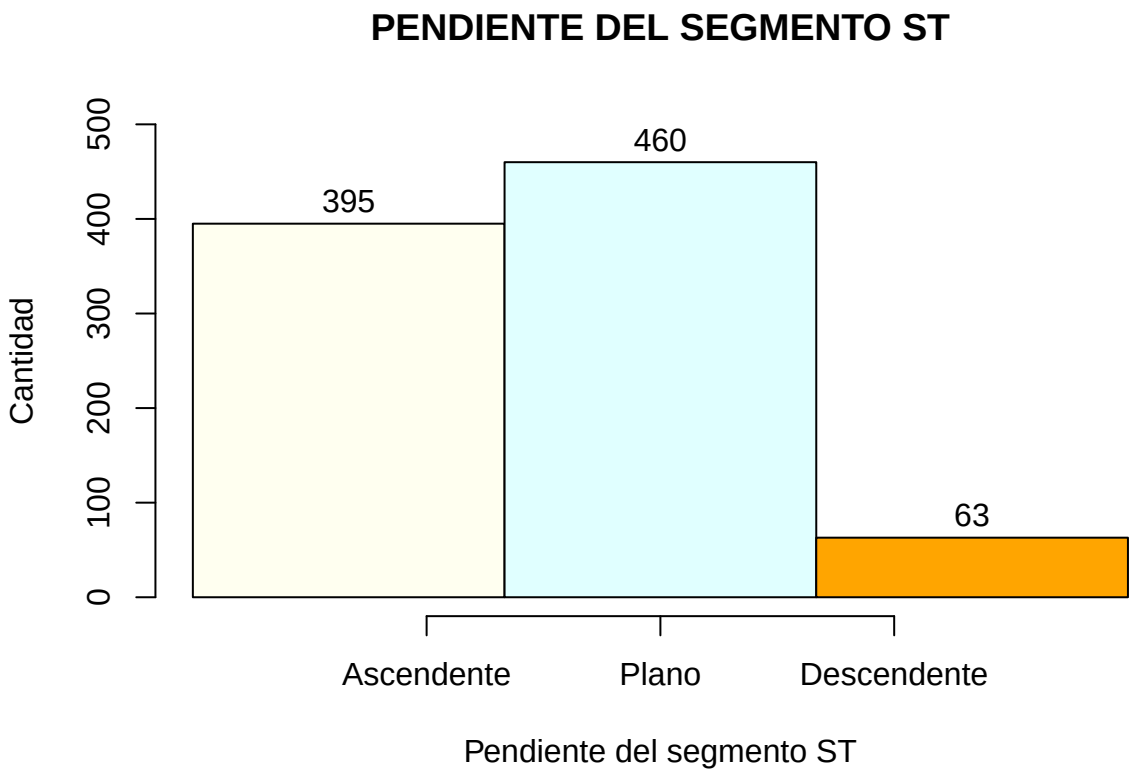
PENDIENTE DEL SEGMENTO ST (ST_Slope)

Como ocurría en otras características anteriores el conjunto tiene los valores para esta caracteristica 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)
```

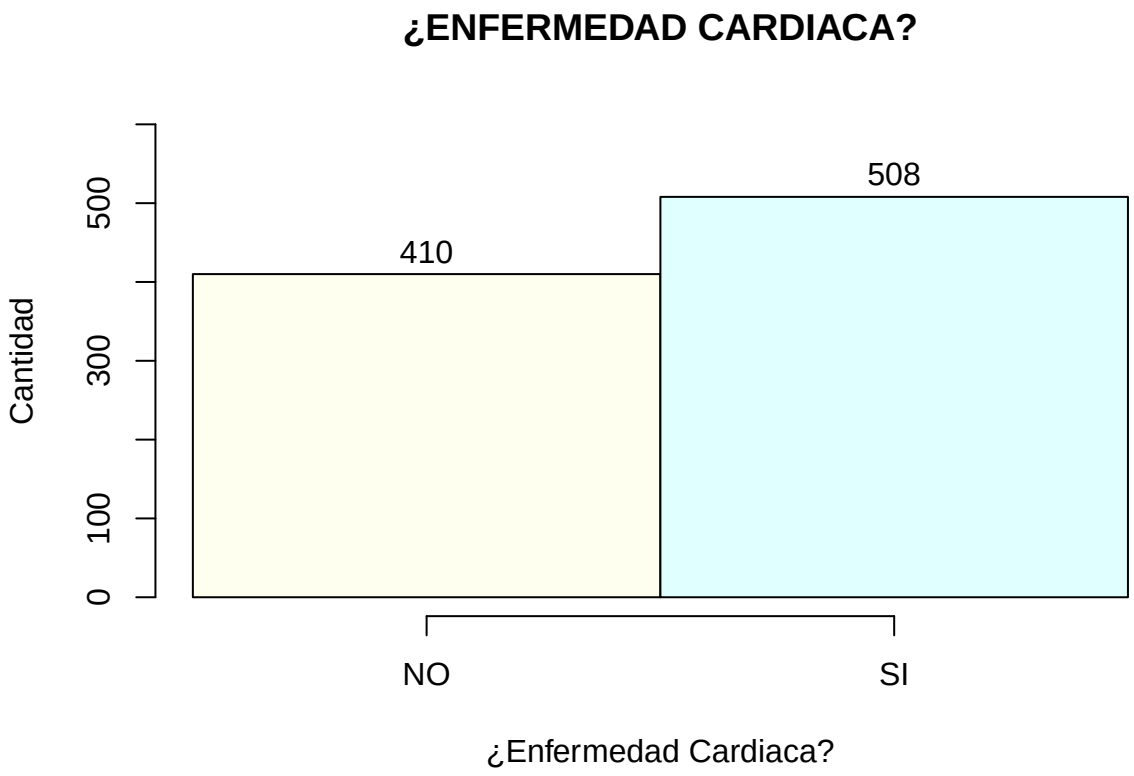


El caso más común es que la pendiente sea plana, teniendo menos casos en los casos descendentes.

¿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.

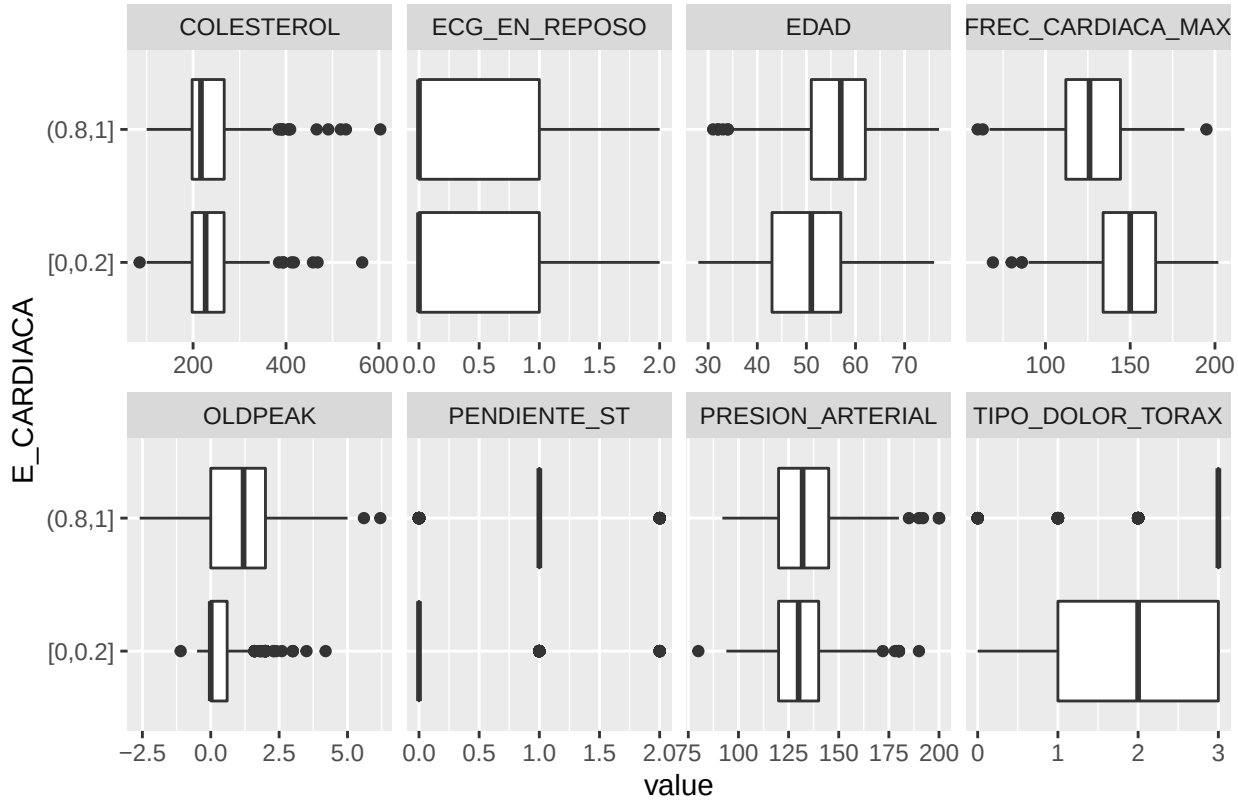
Construcción de conjunto de datos final

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]<-  "PRESION_ARTERIAL"
colnames(datos_final)[5]<-  "COLESTEROL"
colnames(datos_final)[6]<-  "NIVEL_DE_AZUCAR"
colnames(datos_final)[7]<-  "ECG_EN_REPOSO"
colnames(datos_final)[8]<-  "FREC_CARDIACA_MAX"
colnames(datos_final)[9]<-  "ANGINA_x_EJERCICIO"
colnames(datos_final)[10]<- "OLDPEAK"
colnames(datos_final)[11]<- "PENDIENTE_ST"
colnames(datos_final)[12]<- "E_CARDIACA"
```

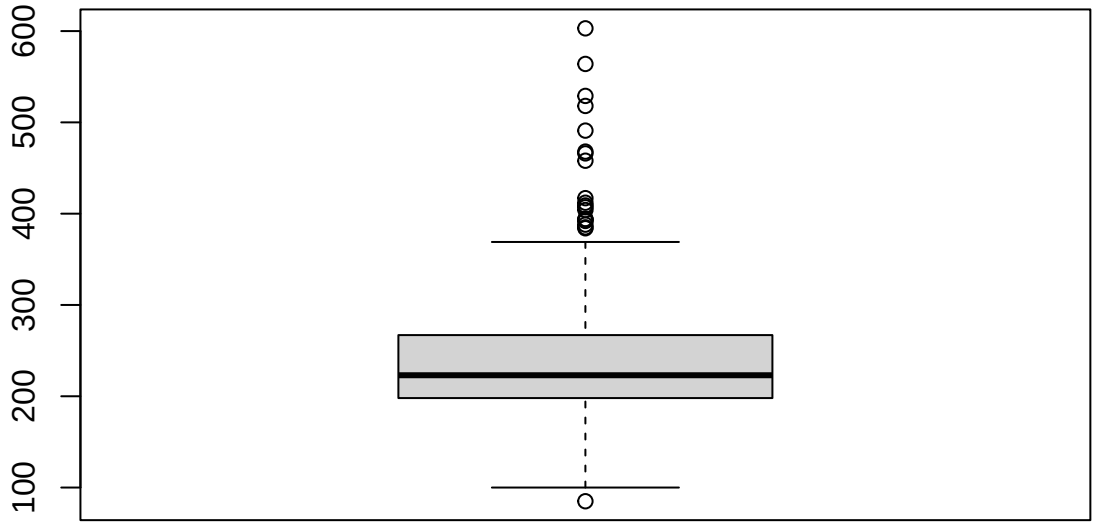
Por ultimo se va a mirar a través de los diagramas de cajas el rango de las características enfrentado a si un paciente tiene una enfermedad cardiaca o no.

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



Eliminamos outliers

```r
datos_bp.colesterol <- boxplot(datos_final$COLESTEROL)
```




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

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

```
datos_bp.colesterol.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.colesterol.out))
dev.off()
```

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

```
datos_frec.cardiaca.max <- boxplot(datos_final$FREC_CARDIACA_MAX)
datos_frec.cardiaca.max.out <- datos_frec.cardiaca.max$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_CARDIACA_MAX %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$PRESION_ARTERIAL)
datos_bp.presion_arterial.out <- datos_bp.presion_arterial$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(!(PRESION_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: "
```

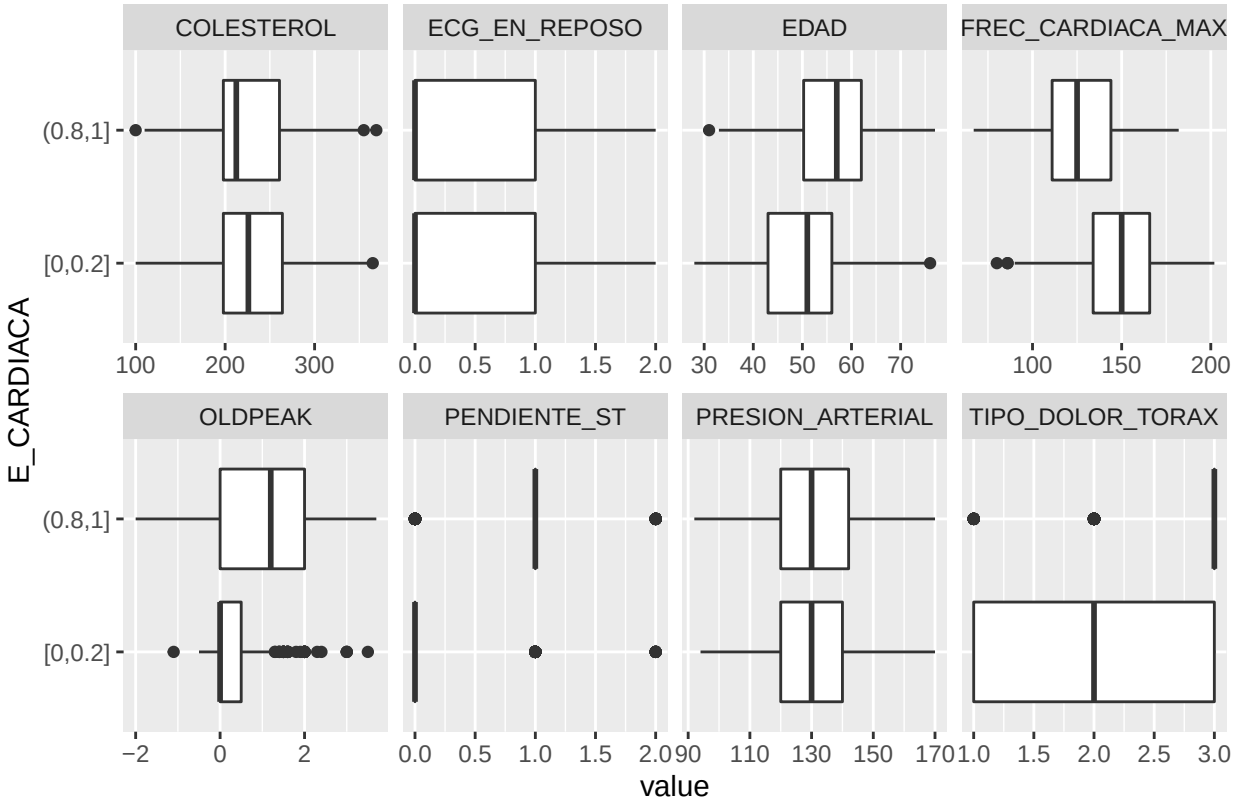
```
datos_bp.tipo_dolor_torax.out
```

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

```
datos_final <- datos_final %>% filter(!(TIPO_DOLOR_TORAX %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")
```



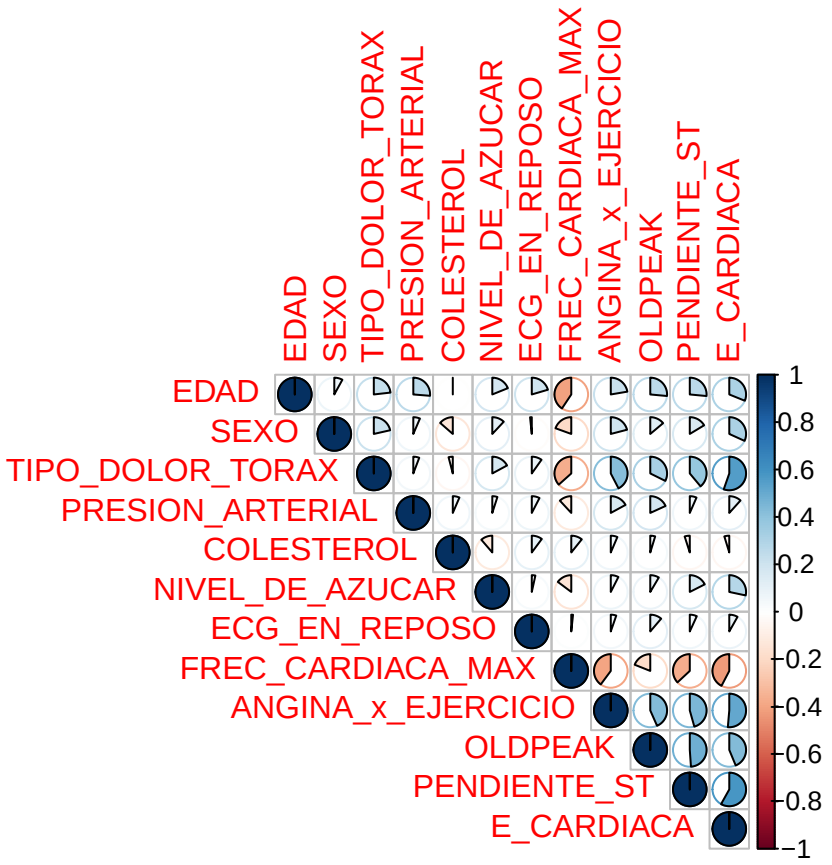
Correlaciones

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

```
##                              EDAD        SEXO TIPO_DOLOR_TORAX PRESION_ARTERIAL
```

```
## EDAD                1.0000000000  0.07987476       0.23222966       0.26263094
## SEXO                0.0798747606  1.00000000       0.21446723       0.06480334
## TIPO_DOLOR_TORAX    0.2322296603  0.21446723       1.00000000       0.05761116
## PRESION_ARTERIAL    0.2626309425  0.06480334       0.05761116       1.00000000
## COLESTEROL         -0.0002226103 -0.13574432      -0.04056882       0.06464654
## NIVEL_DE_AZUCAR     0.1888535829  0.11982017       0.17113944       0.04940893
## ECG_EN_REPOSO       0.2045327265 -0.01651334       0.09834667       0.07584846
## FREC_CARDIACA_MAX  -0.4065414509 -0.19479093      -0.36601006      -0.11900267
## ANGINA_x_EJERCICIO  0.2232404235  0.20949172       0.42401538       0.17055230
## OLDPEAK             0.2647072227  0.13871819       0.32323764       0.18156990
## PENDIENTE_ST        0.2649516503  0.16347763       0.38922582       0.06577099
## E_CARDIACA          0.3112477185  0.31757032       0.55357954       0.11663793
##                       COLESTEROL NIVEL_DE_AZUCAR ECG_EN_REPOSO
## EDAD                -0.0002226103      0.18885358    0.20453273
## SEXO                -0.1357443154      0.11982017   -0.01651334
## TIPO_DOLOR_TORAX    -0.0405688154      0.17113944    0.09834667
## PRESION_ARTERIAL     0.0646465372      0.04940893    0.07584846
## COLESTEROL           1.0000000000     -0.11313552    0.10184968
## NIVEL_DE_AZUCAR     -0.1131355201      1.00000000    0.03428603
## ECG_EN_REPOSO        0.1018496822      0.03428603    1.00000000
## FREC_CARDIACA_MAX    0.1075182058     -0.14779921    0.01659835
## ANGINA_x_EJERCICIO   0.0643936183      0.07501356    0.05301848
## OLDPEAK              0.0471006517      0.08503826    0.11254750
## PENDIENTE_ST        -0.0493791928      0.17525733    0.06863116
## E_CARDIACA          -0.0477984682      0.28341365    0.08182717
##                     FREC_CARDIACA_MAX ANGINA_x_EJERCICIO     OLDPEAK
## EDAD                      -0.40654145         0.22324042  0.26470722
## SEXO                      -0.19479093         0.20949172  0.13871819
## TIPO_DOLOR_TORAX          -0.36601006         0.42401538  0.32323764
## PRESION_ARTERIAL          -0.11900267         0.17055230  0.18156990
## COLESTEROL                 0.10751821         0.06439362  0.04710065
## NIVEL_DE_AZUCAR           -0.14779921         0.07501356  0.08503826
## ECG_EN_REPOSO              0.01659835         0.05301848  0.11254750
## FREC_CARDIACA_MAX          1.00000000        -0.39834669 -0.18975488
## ANGINA_x_EJERCICIO        -0.39834669         1.00000000  0.43365158
## OLDPEAK                   -0.18975488         0.43365158  1.00000000
## PENDIENTE_ST              -0.36930737         0.45642629  0.48875764
## E_CARDIACA                -0.42346641         0.51376813  0.43751373
##                     PENDIENTE_ST  E_CARDIACA
## EDAD                  0.26495165  0.31124772
## SEXO                  0.16347763  0.31757032
## TIPO_DOLOR_TORAX      0.38922582  0.55357954
## PRESION_ARTERIAL      0.06577099  0.11663793
## COLESTEROL           -0.04937919 -0.04779847
## NIVEL_DE_AZUCAR       0.17525733  0.28341365
## ECG_EN_REPOSO         0.06863116  0.08182717
## FREC_CARDIACA_MAX    -0.36930737 -0.42346641
## ANGINA_x_EJERCICIO    0.45642629  0.51376813
## OLDPEAK               0.48875764  0.43751373
## PENDIENTE_ST          1.00000000  0.58120599
## E_CARDIACA            0.58120599  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 caracteristicas

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

```
##              EDAD              SEXO    TIPO_DOLOR_TORAX   PRESION_ARTERIAL
##        88.1377792         0.1621106         0.6374466       245.0442666
##       COLESTEROL    NIVEL_DE_AZUCAR      ECG_EN_REPOSO  FREC_CARDIACA_MAX
##       2160.3249895         0.1754691         0.6335349       637.4350996
## ANGINA_x_EJERCICIO           OLDPEAK      PENDIENTE_ST         E_CARDIACA
##         0.2422160         0.9478357         0.3549759         0.2478786
```

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

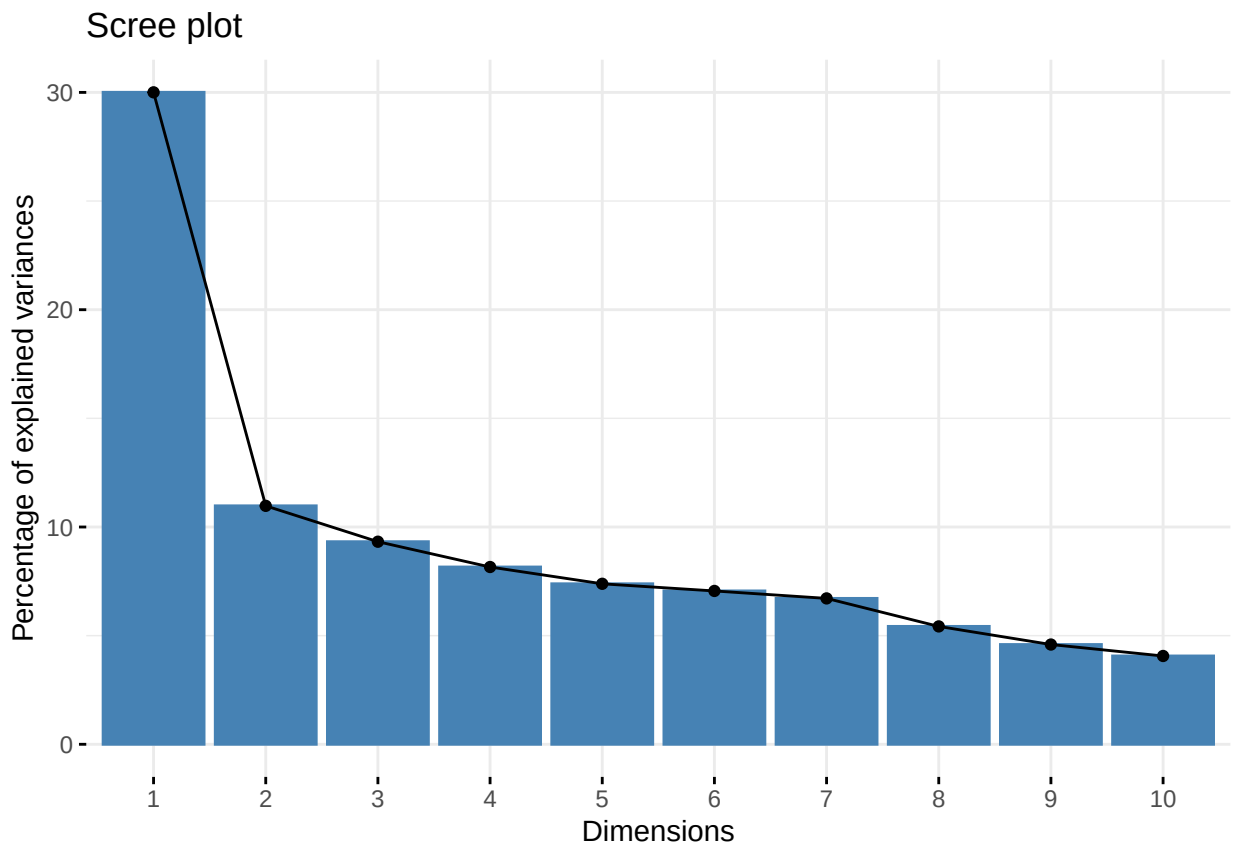
Lo siguiente es centrar y escalar las características, para que así las variables pierdan esa variabilidad. Una vez calculada la matriz se la asigno al pca

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

```
## Standard deviations (1, .., p=12):
##  [1] 1.8974493 1.1475086 1.0577031 0.9894707 0.9414635 0.9202567 0.8976197
##  [8] 0.8068512 0.7422484 0.6983699 0.6456073 0.5828417
##
## Rotation (n x k) = (12 x 12):
##                             PC1         PC2         PC3         PC4          PC5
## EDAD                 0.28014843 -0.25245541  0.48732200 -0.07304964  0.256104929
## SEXO                 0.20039780  0.33847469  0.03514677 -0.16476622 -0.763305970
## TIPO_DOLOR_TORAX     0.35837588  0.09136430 -0.13473172  0.16514131  0.005882123
## PRESION_ARTERIAL     0.13829428 -0.37958423  0.36362321 -0.59017736 -0.255760964
## COLESTEROL          -0.03058803 -0.57356528 -0.32677314  0.01280796 -0.005505378
## NIVEL_DE_AZUCAR      0.17188118  0.22329356  0.46905923  0.37182111  0.024273900
## ECG_EN_REPOSO        0.08311302 -0.46159376  0.23741130  0.58968414 -0.321590348
## FREC_CARDIACA_MAX   -0.33268947 -0.13119755 -0.16819561  0.23310113 -0.385916552
## ANGINA_x_EJERCICIO   0.37320411 -0.07541656 -0.29323460 -0.15482224  0.002822528
## OLDPEAK              0.33368710 -0.20542115 -0.24820847  0.01026826 -0.089164578
## PENDIENTE_ST         0.38478714  0.03185974 -0.20887652  0.12195037  0.151652559
## E_CARDIACA           0.43248195  0.10606714 -0.09073580  0.11184415 -0.048966672
##                             PC6         PC7         PC8         PC9         PC10
## EDAD                -0.19012829  0.07602767 -0.39586282  0.38465564  0.152868605
## SEXO                -0.11787912  0.27816619 -0.34470023  0.04446174 -0.013857771
## TIPO_DOLOR_TORAX    -0.10759466  0.20321763  0.60091060  0.54901398  0.003766156
## PRESION_ARTERIAL     0.22105853 -0.19709636  0.37321438 -0.07017404 -0.234844352
## COLESTEROL           0.33735336  0.62122599 -0.19252011  0.05017177 -0.123957417
## NIVEL_DE_AZUCAR      0.68485907  0.11737352  0.02332883 -0.16064936  0.172660255
## ECG_EN_REPOSO       -0.40796143 -0.05485229  0.10908375 -0.27682556 -0.025367300
## FREC_CARDIACA_MAX    0.31452613 -0.32365282  0.06117644  0.29035242  0.035570020
## ANGINA_x_EJERCICIO  -0.01691429  0.05956734  0.17454230 -0.48906281  0.549950590
## OLDPEAK              0.16611307 -0.49474050 -0.28538178  0.26579163  0.355664752
## PENDIENTE_ST         0.07037292 -0.27927242 -0.23036132 -0.20690027 -0.608826995
## E_CARDIACA           0.09209677  0.05414122  0.07660088  0.04271721 -0.273082385
##                            PC11        PC12
## EDAD                -0.41570328 -0.08124084
## SEXO                 0.02697977 -0.14584048
## TIPO_DOLOR_TORAX     0.08643164 -0.30804525
## PRESION_ARTERIAL     0.05681729 -0.02320323
## COLESTEROL           0.08777809 -0.02704705
## NIVEL_DE_AZUCAR      0.10225997 -0.11095277
## ECG_EN_REPOSO        0.11546071  0.02374709
## FREC_CARDIACA_MAX   -0.58343555 -0.08461831
## ANGINA_x_EJERCICIO  -0.38969530 -0.13144093
## OLDPEAK              0.46313399  0.10654434
## PENDIENTE_ST        -0.12372310 -0.45235695
## E_CARDIACA          -0.24942136  0.78909996
```

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.

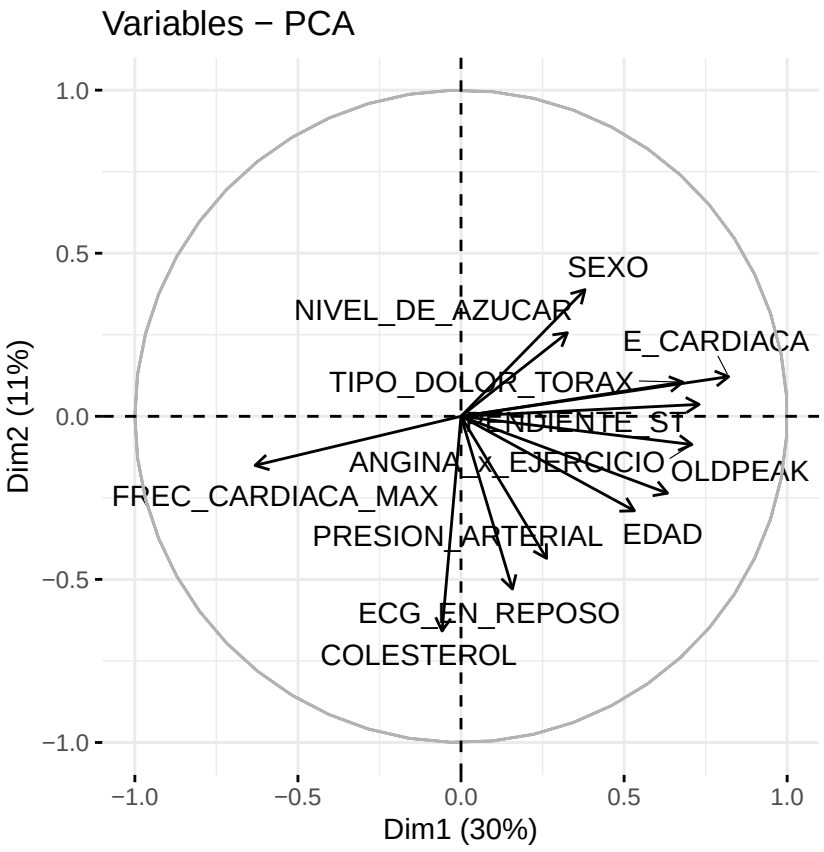
Para confirmar la interpretación, no estaría de más obtener las estadísticas de todas las componentes

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

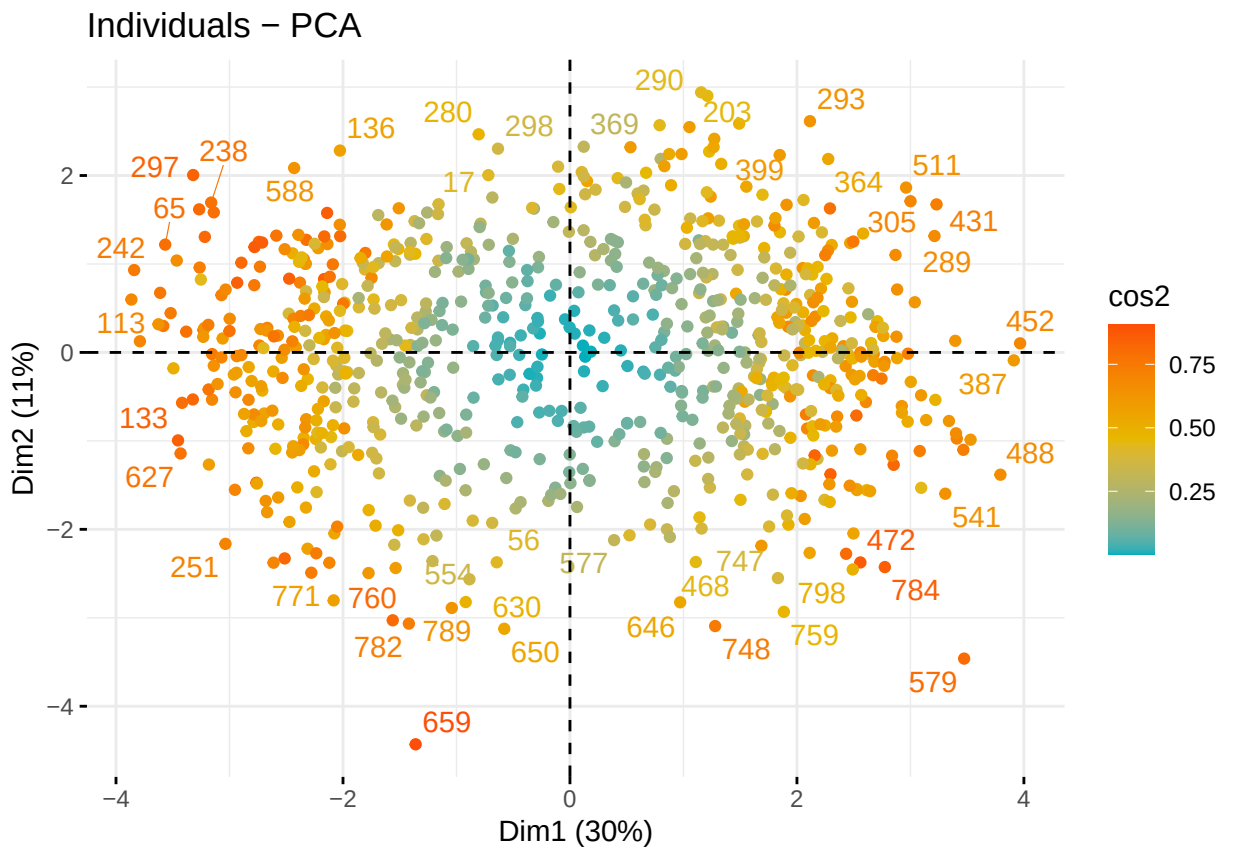
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.897 1.1475 1.05770 0.98947 0.94146 0.92026 0.89762
## Proportion of Variance 0.300 0.1097 0.09323 0.08159 0.07386 0.07057 0.06714
## Cumulative Proportion  0.300 0.4098 0.50299 0.58457 0.65844 0.72901 0.79615
##                           PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.80685 0.74225 0.69837 0.64561 0.58284
## Proportion of Variance 0.05425 0.04591 0.04064 0.03473 0.02831
## Cumulative Proportion  0.85040 0.89631 0.93696 0.97169 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.

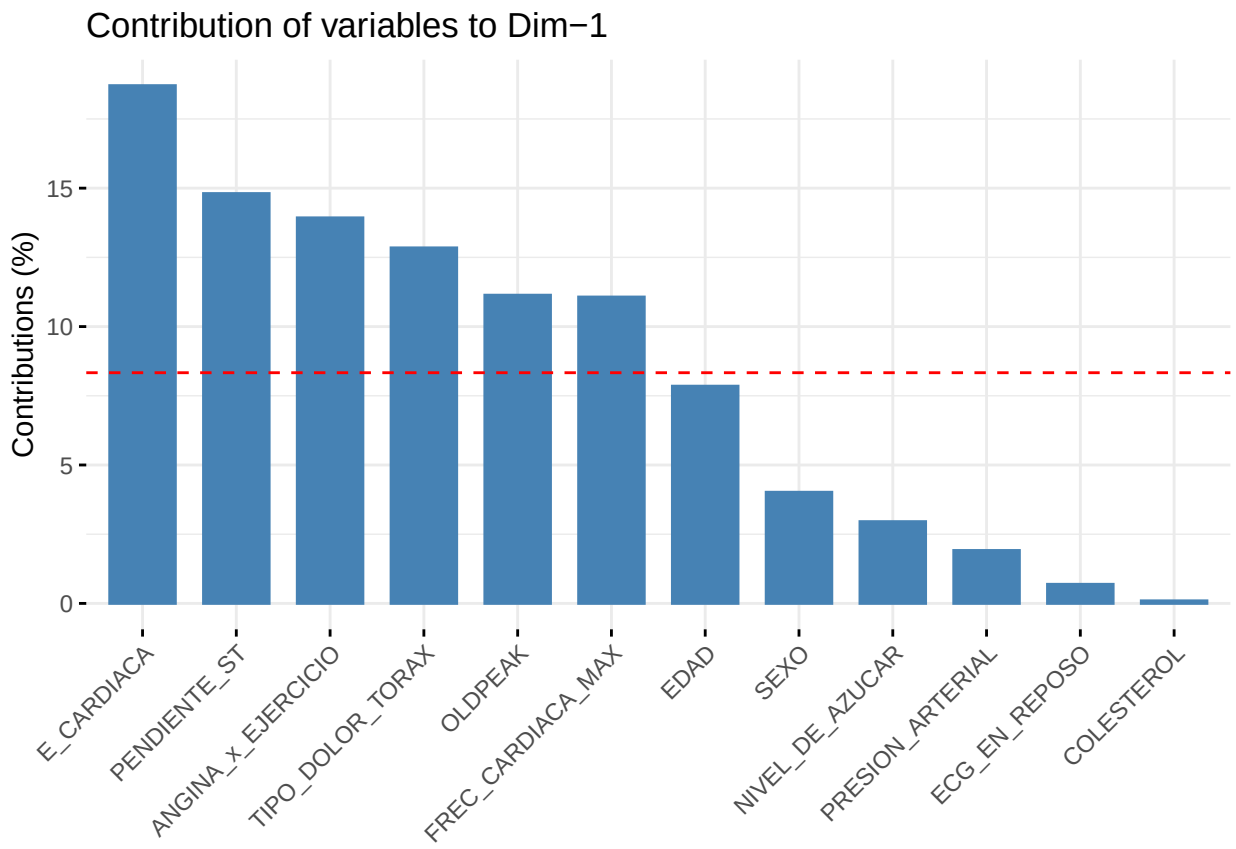
```
#Representación de variables sobre componentes principales
fviz_pca_var(pca, repel = TRUE, scale = 0)
```




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

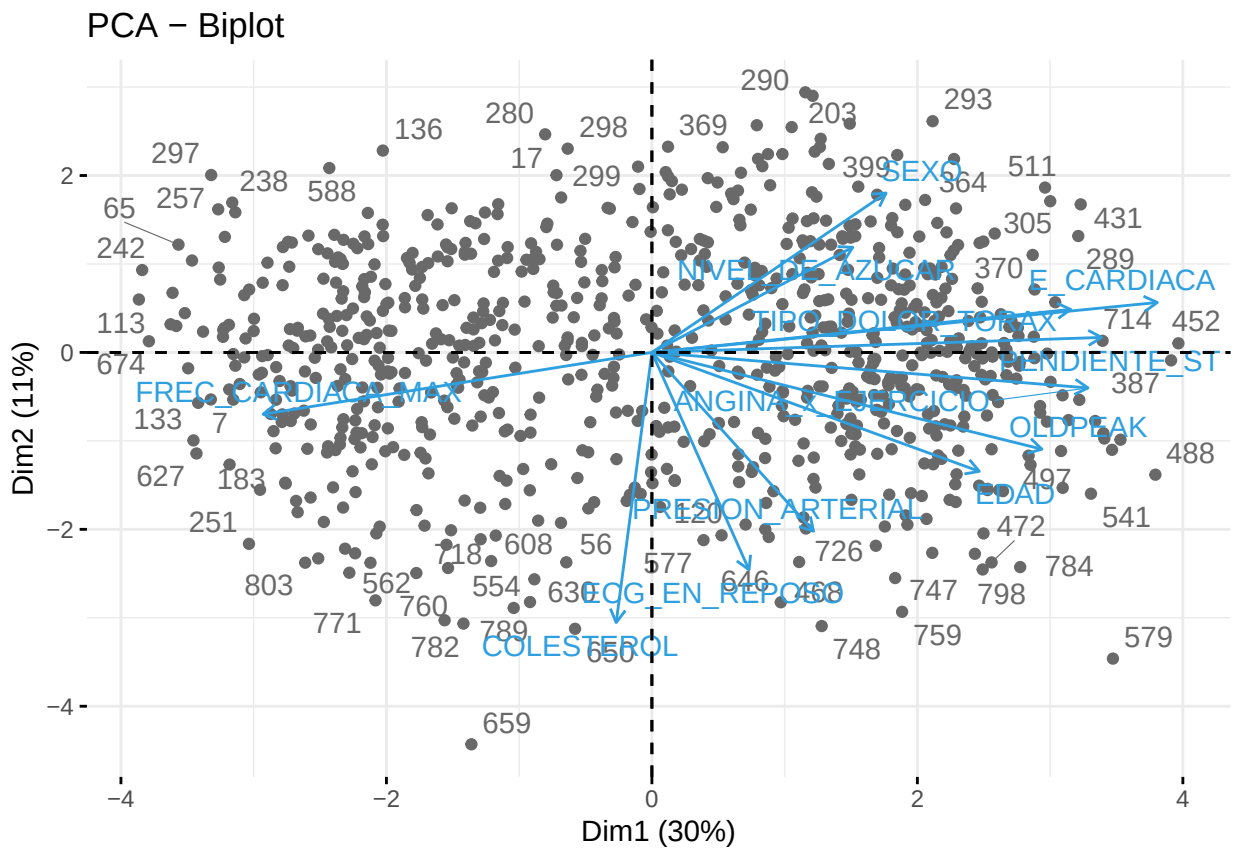



```r
#Representa la contribución de filas/columnas de los resultados de un pca
fviz_contrib(pca,choice = "var")
```



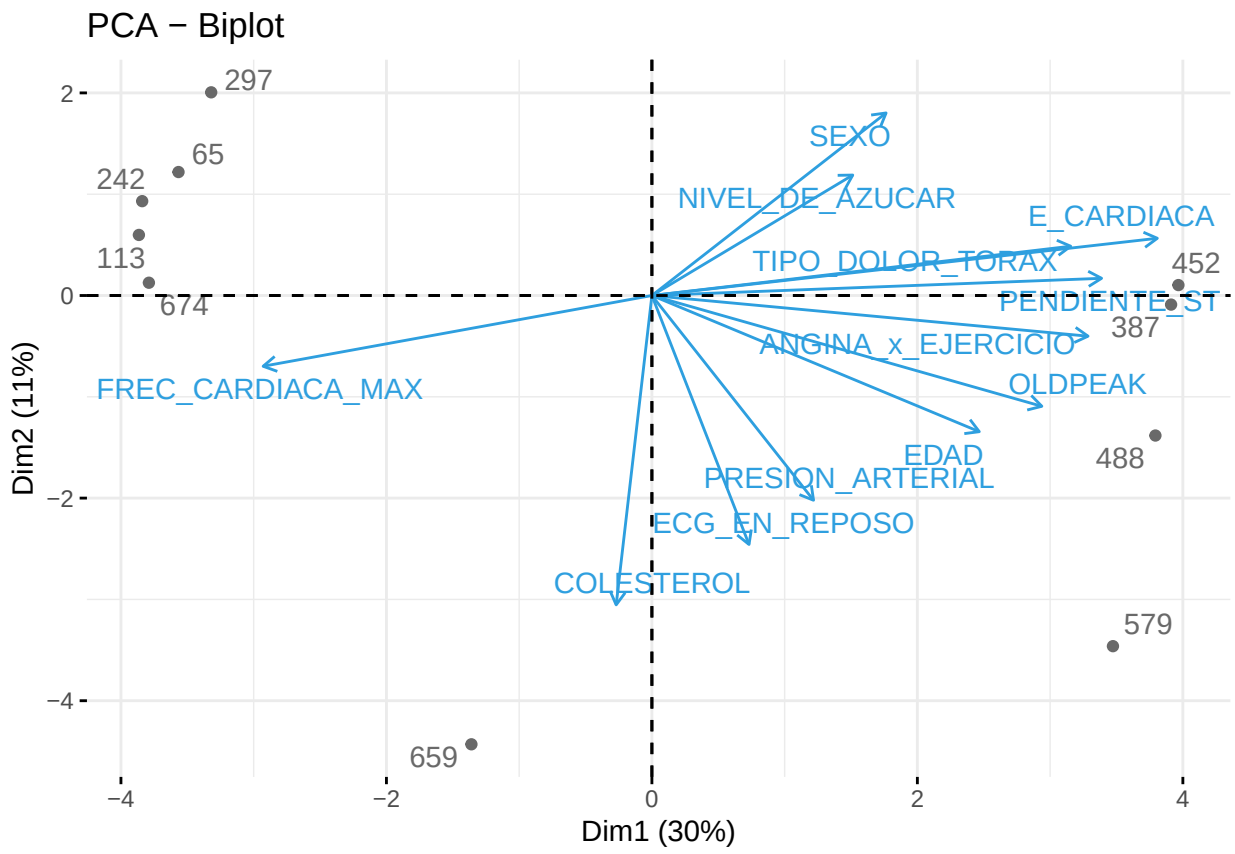
Una vez que hemos representada las variables y los individuos, se va a fusionar estas dos gráficas

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

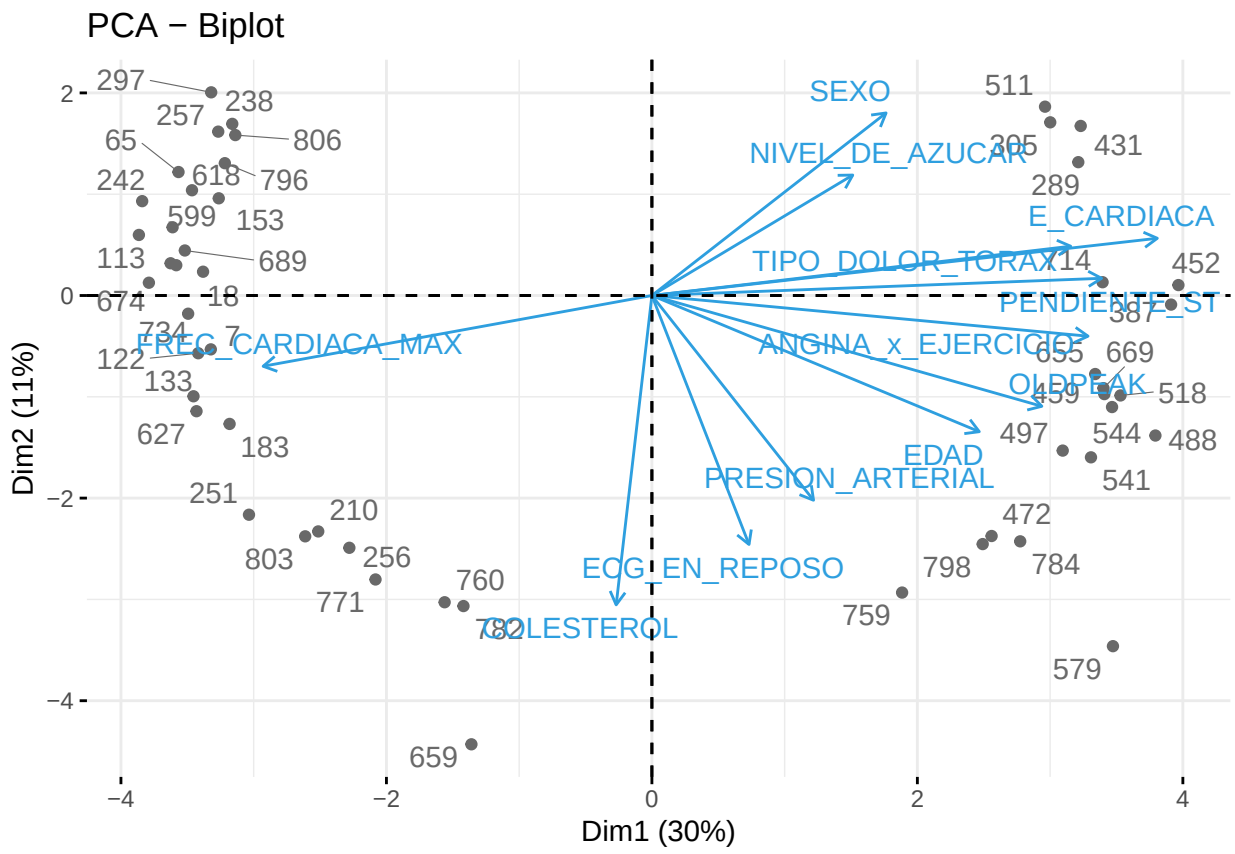


Aunque la opción de repelerse esta activada al ser bastantes casos no se puede ver una manera correcta, así que se a mostrar solamente los 10, 50 y 100 casos más influyentes

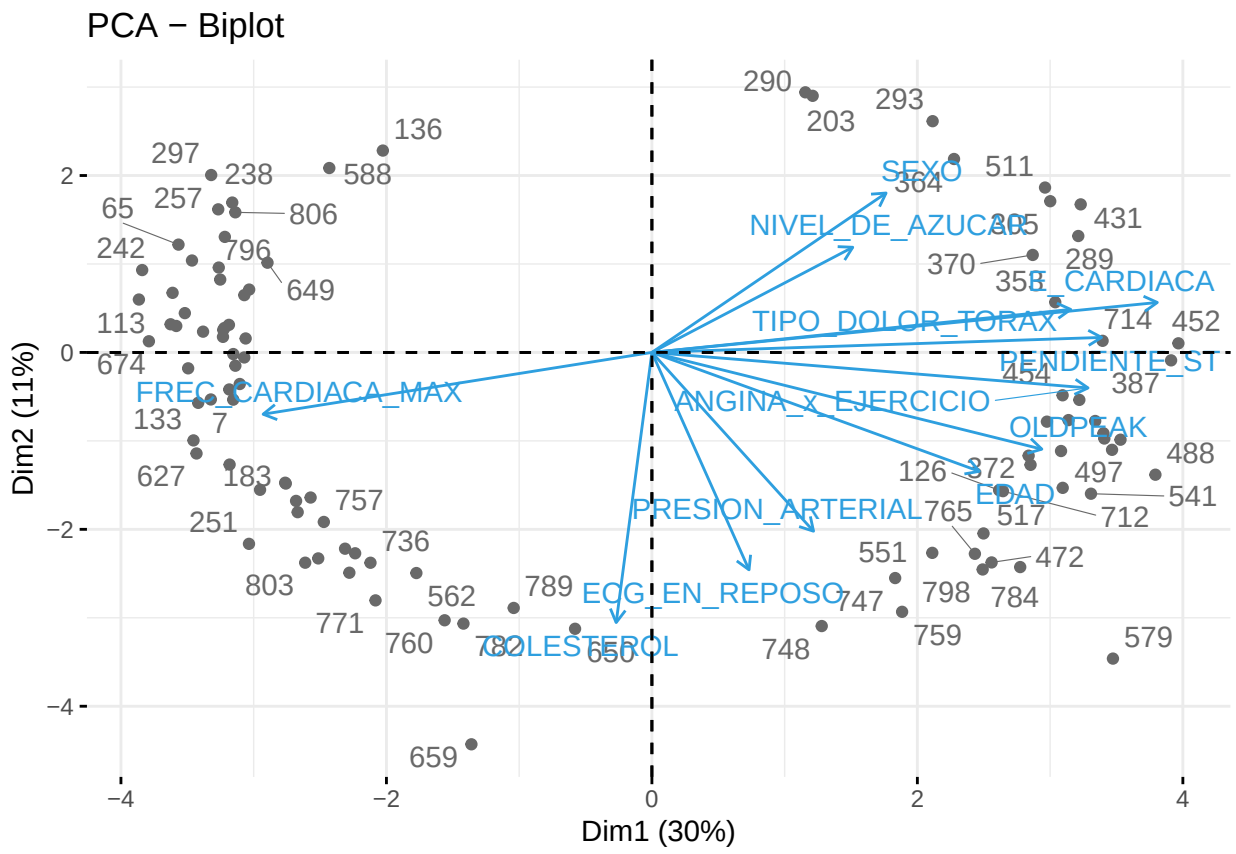
```
#Representación de variables y los 10 individuos más influyentes en la misma gráfica
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF",
                col.ind = "#696969", select.ind = list(contrib = 10))
```




```
#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))
```




```
#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 variables numericas

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

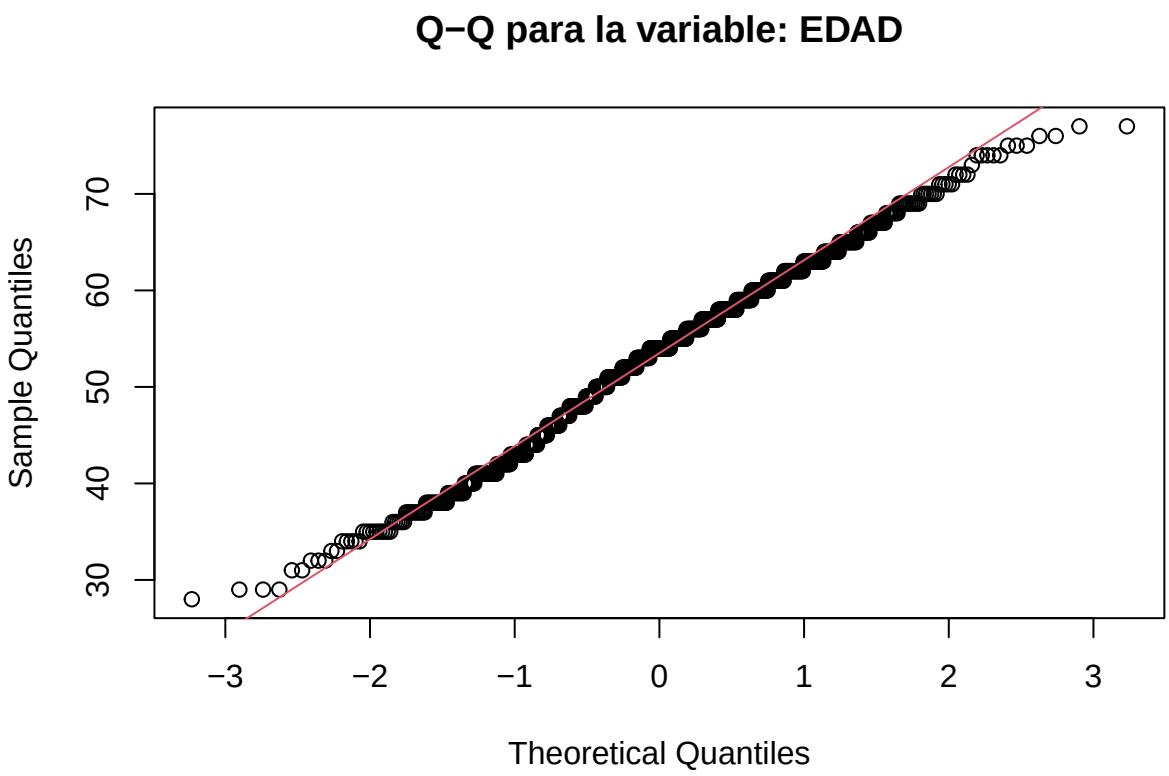
Podemos comprobar que para cada prueba se obtiene un p-valor superior al nivel de significancia estadistica prefijado en alpha = 0,05. Si esto se cumple, entonces se considera que variable en cuestion sigue la distribución normal.

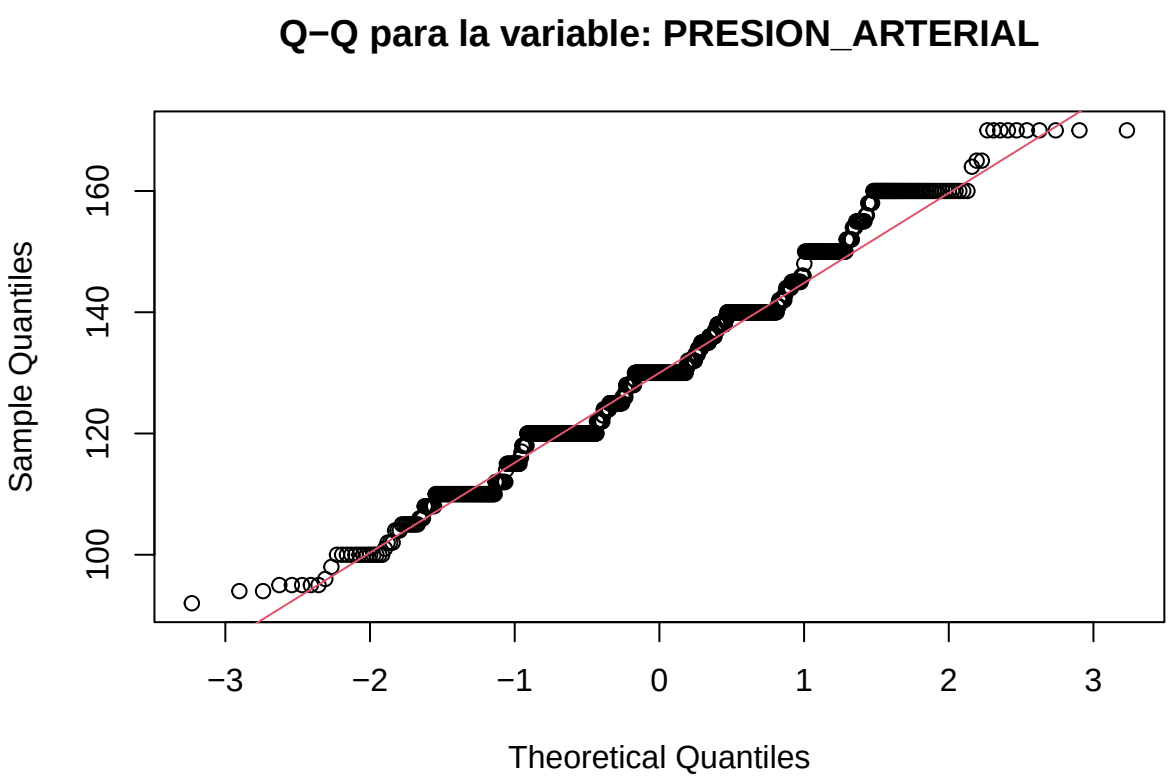
```r
if (!require('nortest')) install.packages('nortest'); library('nortest')
alpha = 0.05
col.names = colnames(datos_final)
ind = 1

# Comprobanos unicamente lS 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
  }
```

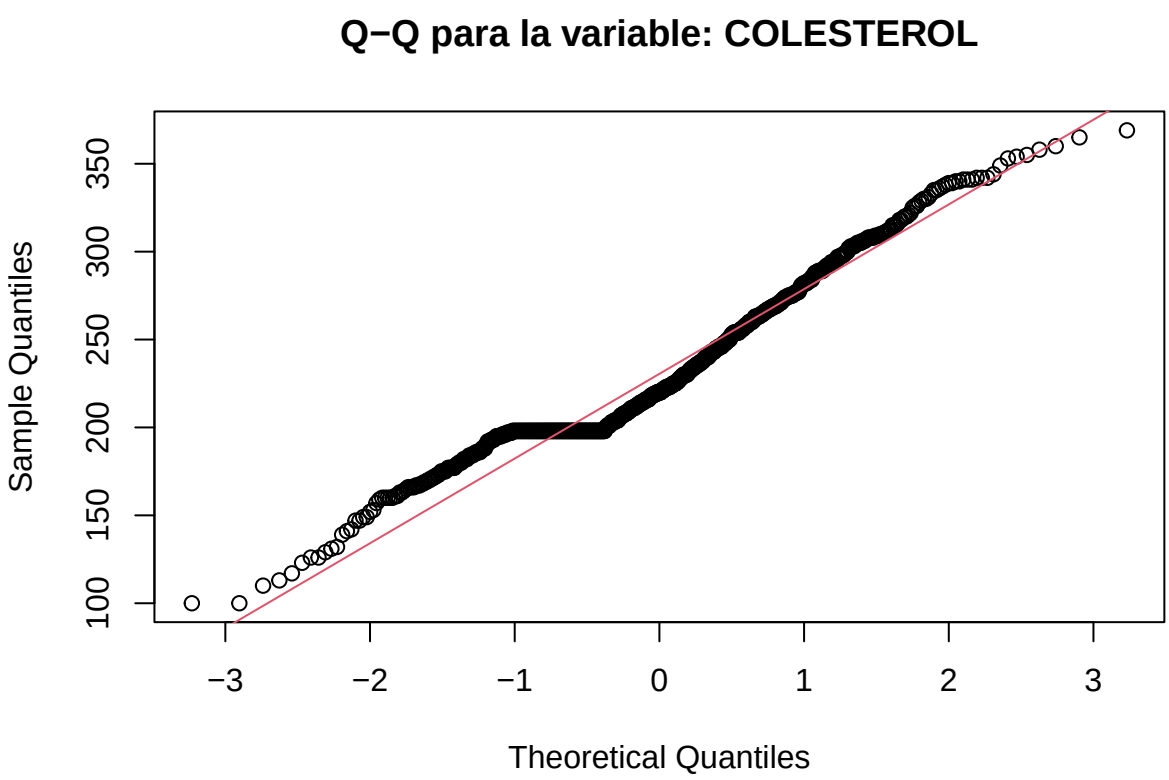
```
## Variables que no siguen una distribución normal:
## EDAD, PRESION_ARTERIAL, COLESTEROL, NIVEL_DE_AZUCAR,
## FREC_CARDIACA_MAX, OLDPEAK, E_CARDIACA
```

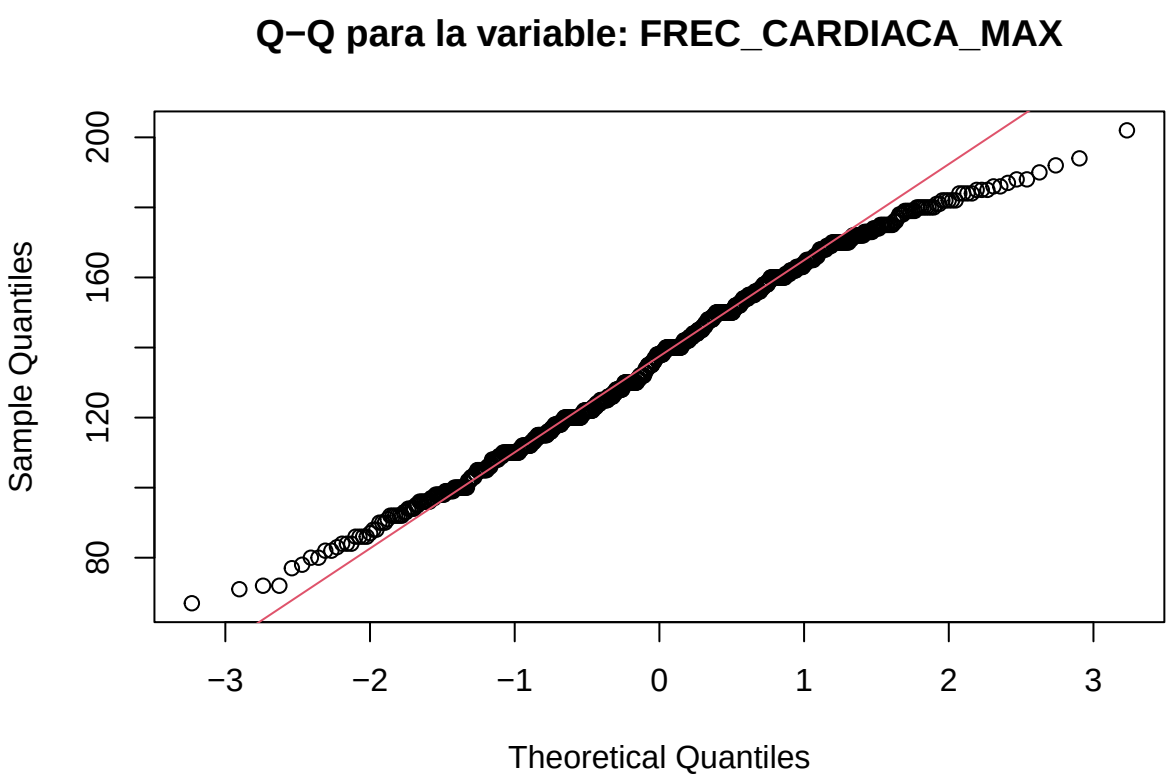
Podemos realizar un Q-Q plot para comprobar si las variables obtenidas en el anterior punto no siguen una distribución normal.

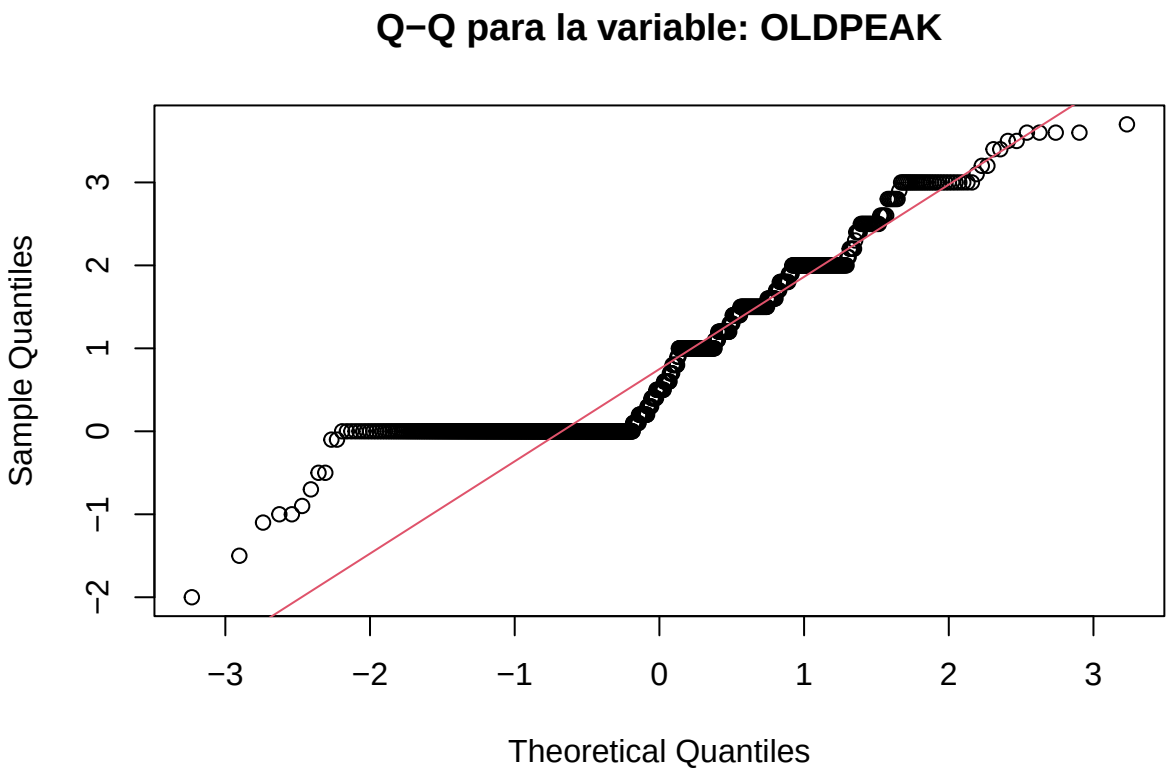
```r
variables <- c("EDAD", "PRESION_ARTERIAL", "COLESTEROL",
               "FREC_CARDIACA_MAX", "OLDPEAK")


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
Sample Quantiles
30
40
50
60
70
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Q−Q para la variable: PRESION_ARTERIAL
Sample Quantiles
100
120
140
160
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Q−Q para la variable: COLESTEROL
Sample Quantiles
100
150
200
250
300
350
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Q−Q para la variable: FREC_CARDIACA_MAX
Sample Quantiles
80
120
160
200
−3
−2
−1
0
1
2
3
Theoretical Quantiles



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

Pruebas estadisticas

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

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

# Calculamos el coficiente 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

    # Aañade 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                0.32133836 5.839139e-21
## PRESION_ARTERIAL    0.11079517 1.566612e-03
## COLESTEROL         -0.06760546 5.414249e-02
## NIVEL_DE_AZUCAR     0.28341365 1.825875e-16
## FREC_CARDIACA_MAX -0.42471804 6.745047e-37
## OLDPEAK             0.44538146 8.078414e-41
```

Podemos identificar cual es la variable más correlacionada con la variable Enfermedad Cardiaca, viendo cuales de los valores de la columan 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 varibale objetivo son: OLDPEAK y FREC CARDÍACA MAX.

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

Grupos de datos

Podemos establecer una serie de grupos partiendo de nuestro dataset para analizar y/o comparlos 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$NIVEL_DE_AZUCAR == 0,]
datos_final.con_hiperglucemia <- datos_final[datos_final$NIVEL_DE_AZUCAR == 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]
```

```r
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 estadistica 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.

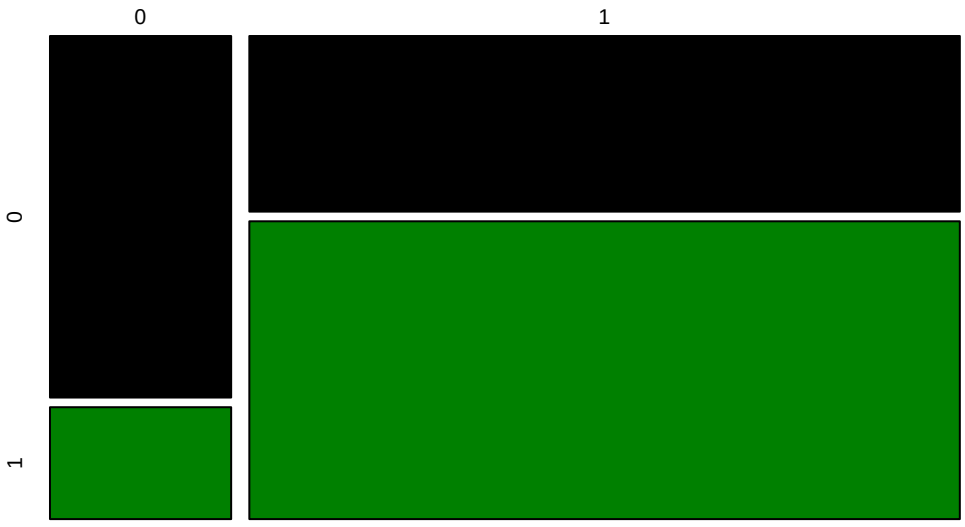
Test estadísticos de significancia

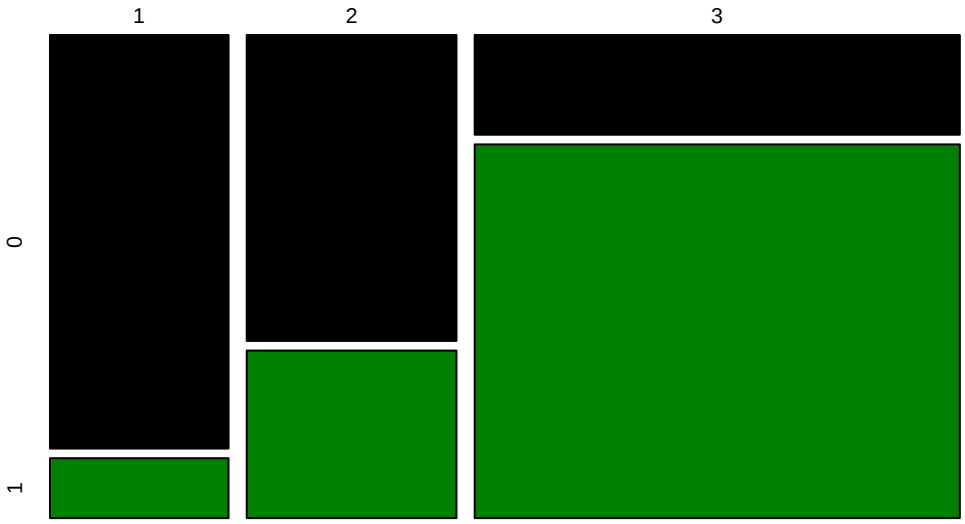
Antes de proceder a la clasificación de los parámetros de pacientes con más probabilidades de sufrir enfermedad cardiaca, deberemos de hacer una selección previa de las caracteristicas a utilizar en nuestro modelo.

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.

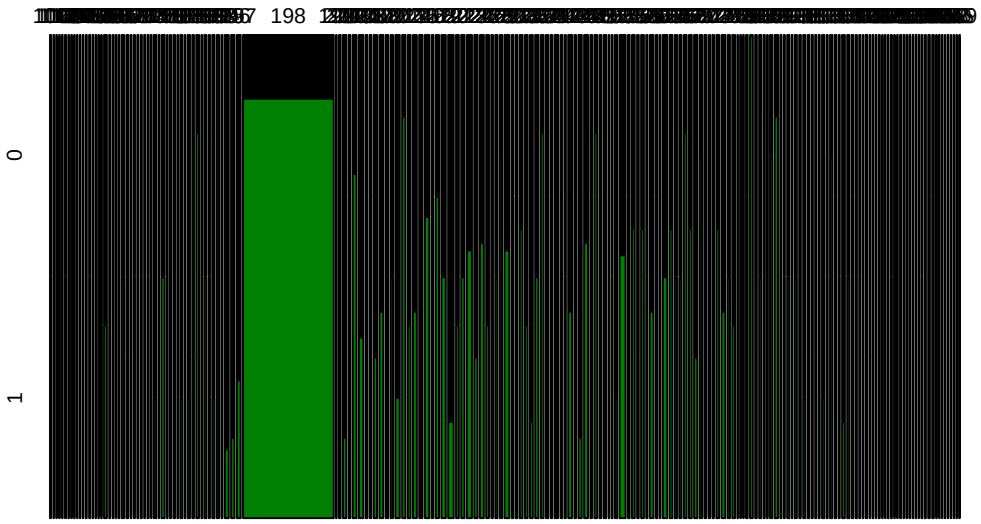
```r
if(!require("DescTools"))
{
  install.packages("DeskTools")
  # http://cran.us.r-project.org
  library("DescTools")
}

# Analizamos las correlaciones de todos las caracteristicas 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)
  }

  vector_tipos[5] <- 'character'
```

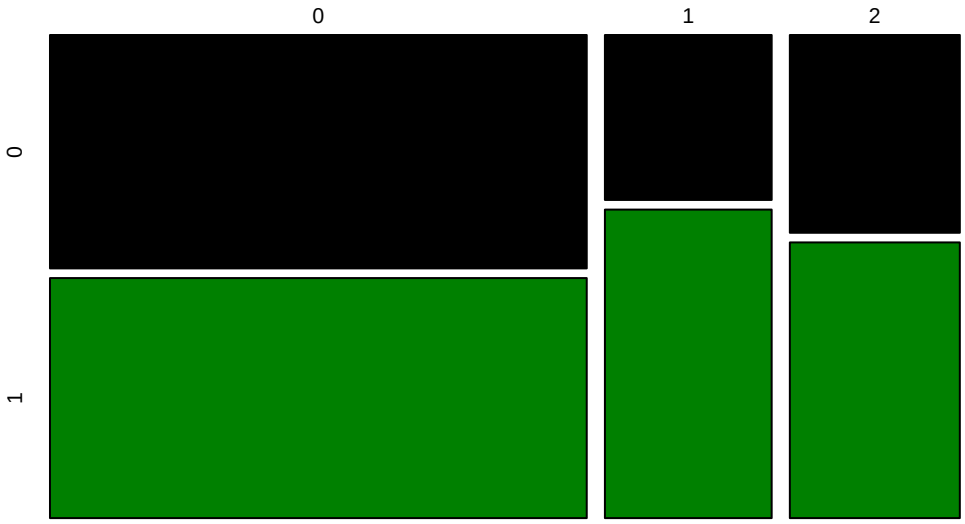
```
  if (vector_tipos[ind] != 'numeric')
  {
    # Solo pintamos las variables categoricas ya que con las de tipo numerico no se aprecian los valores
    plot(tabla_cruzada, col = c("black","#008000"), main = paste0(i, " vs. E. CARDIACA"))
  }


  ind <- ind + 1

}
```

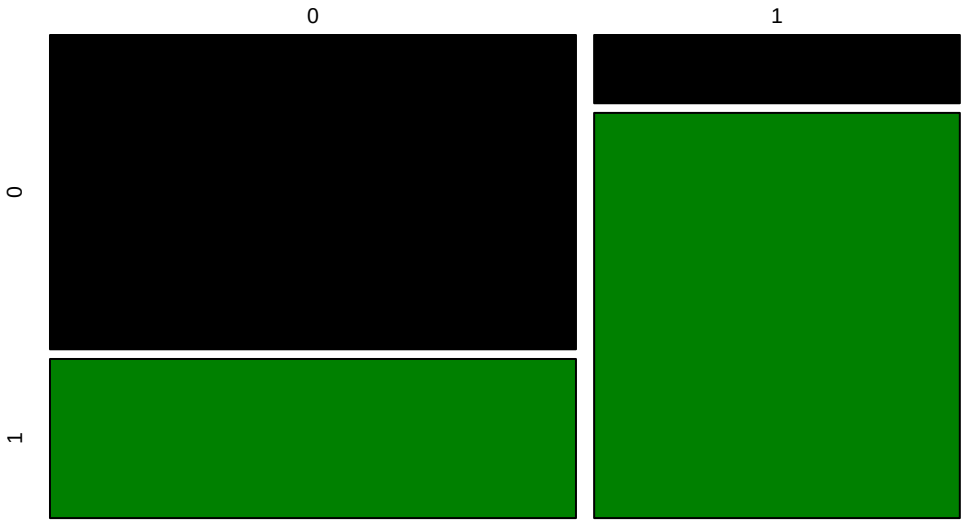


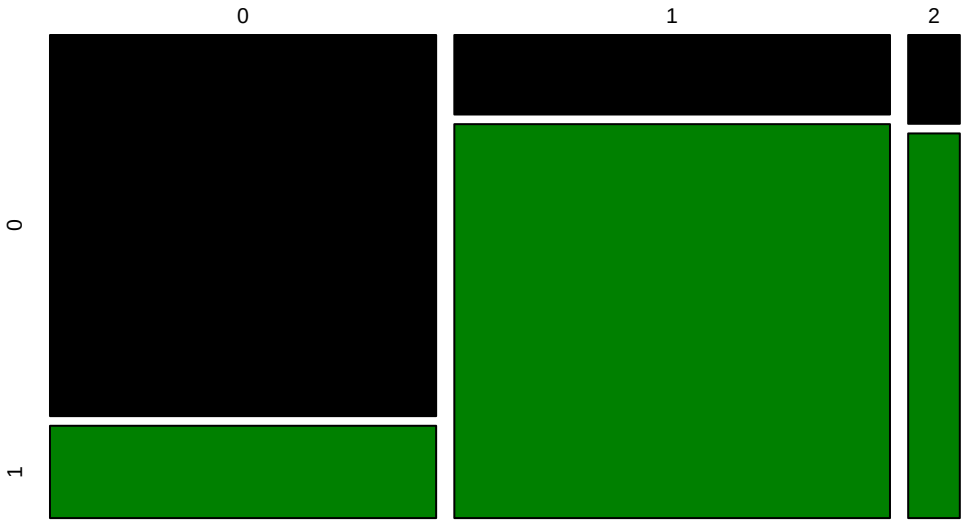

TIPO_DOLOR_TORAX vs. E. CARDIACA
1
2
3
0
1

COLESTEROL vs. E. CARDIACA




ECG_EN_REPOSO vs. E. CARDIACA
0
1
2
0
1


ANGINA_x_EJERCICIO vs. E. CARDIACA
0
1
0
1




```
n_list <- list(nombre=as.character(datos_corr.nombre),
               CamerV=as.numeric(datos_corr.CramerV),
               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 0.633914979247456 0.633914979247456
## 5          COLESTEROL 0.600808356388246 0.600808356388246
## 3    TIPO_DOLOR_TORAX 0.560309860170942 0.560309860170942
## 8   FREC_CARDIACA_MAX 0.530074780053706 0.530074780053706
## 10            OLDPEAK 0.520727759286187 0.520727759286187
## 9  ANGINA_x_EJERCICIO 0.513768132124416 0.513768132124416
## 1                EDAD 0.404380273871888 0.404380273871888
## 4    PRESION_ARTERIAL 0.331282489356957 0.331282489356957
## 2                SEXO 0.317570321818606 0.317570321818606
## 6     NIVEL_DE_AZUCAR 0.283413646865563 0.283413646865563
## 7       ECG_EN_REPOSO 0.115604279173622 0.115604279173622
```

Obtenemos las siguientes conclusiones del dataset:

En cuanto al Sexo, las Mujeres tienen menos probabilidad de sufrir una enfermedad cardiaca.

En cuanto al tipo de dolor de Torax, los pacientes que sufren de dolor tipo Asintomáticos son los que pese a lo que se pdoría pensar tienen más probabilidades de sufrir enfermedad cardiaca.

En cuanto a la variable ECG en Reposo está muy equilibrada, pero tienen una ligera mayor probabilidad de sufrir enfermedad cardiaca los de tipo 1 ()

Valores de la V de Cramér (https://en.wikipedia.org/wiki/CramÃĺr%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 podria utilizar un número mayor de variables, pero podría hacerse mucho complejo (con muchas reglas de decisión)

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

```
## [1] "TIPO_DOLOR_TORAX"   "COLESTEROL"         "FREC_CARDIACA_MAX"
## [4] "ANGINA_x_EJERCICIO" "OLDPEAK"            "PENDIENTE_ST"
```

Aplicación del modelo Decision Tree (Arbol de decisión)

Aplicamos el modelo Decision Tree sobre las 3 caracteristicas que hemos obtenido en el test de significacia estadistica de Cramer V.

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


# Reducimos el datset
for(i in colnames(datos_final))
{
  if(!i %in% nombres_columnas)
  {
    if (!i == "E_CARDIACA")
    {
      datos_final$i <- NULL
    }
  }
}
```

Para aplicar los modelos de arbol de decision debemos de discretizar las variables COLESTEROL y FREC CARDIACA MAXIMA.

```
datos_final <- datos_final %>% mutate(COLESTEROL = case_when(
  COLESTEROL < 100 ~ 0,
  (COLESTEROL >= 100 & COLESTEROL < 200) ~ 1,
  (COLESTEROL >= 200 & COLESTEROL < 300) ~ 2,
  (COLESTEROL >= 300 & COLESTEROL < 400) ~ 3,
  (COLESTEROL >= 400 & COLESTEROL < 500) ~ 4,
  COLESTEROL >= 600 ~ 5,
  ))
```

```
datos_final <- datos_final %>% mutate(FREC_CARDIACA_MAX = case_when(
  FREC_CARDIACA_MAX < 50 ~ 0,
  (FREC_CARDIACA_MAX >= 50 & FREC_CARDIACA_MAX < 80) ~ 1,
  (FREC_CARDIACA_MAX >= 80 & FREC_CARDIACA_MAX < 110) ~ 2,
  (FREC_CARDIACA_MAX >= 110 & FREC_CARDIACA_MAX < 140) ~ 3,
  (FREC_CARDIACA_MAX >= 140 & FREC_CARDIACA_MAX < 170) ~ 4,
  (FREC_CARDIACA_MAX >= 170 & FREC_CARDIACA_MAX < 200) ~ 5,
  FREC_CARDIACA_MAX >= 200 ~ 6,
  ))

datos_final
```

```
## # A tibble: 812 x 12
##     EDAD  SEXO TIPO_DO~1 PRESI~2 COLES~3 NIVEL~4 ECG_E~5 FREC_~6 ANGIN~7 OLDPEAK
##    <dbl> <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1    40     1         1     140       2       0       0       5       0     0
##  2    49     0         2     160       1       0       0       4       0     1
##  3    37     1         1     130       2       0       1       2       0     0
##  4    48     0         3     138       2       0       0       2       1     1.5
##  5    54     1         2     150       1       0       0       3       0     0
##  6    39     1         2     120       3       0       0       5       0     0
##  7    45     0         1     130       2       0       0       5       0     0
##  8    54     1         1     110       2       0       0       4       0     0
##  9    37     1         3     140       2       0       0       3       1     1.5
## 10    48     0         1     120       2       0       0       3       0     0
## # ... with 802 more rows, 2 more variables: PENDIENTE_ST <dbl>,
## #   E_CARDIACA <dbl>, and abbreviated variable names 1: TIPO_DOLOR_TORAX,
## #   2: PRESION_ARTERIAL, 3: COLESTEROL, 4: NIVEL_DE_AZUCAR, 5: ECG_EN_REPOSO,
## #   6: FREC_CARDIACA_MAX, 7: ANGINA_x_EJERCICIO
```

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

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

```
set.seed(666)
y <- datos_final$E_CARDIACA # Variable objetivo
X <- datos_final[nombres_columnas] # Variables predictoras

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
## 366 446
```

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

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

```
summary(train_y)
```

```
##   0   1
## 249 292
```

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

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

```
summary(test_y)
```

```
##   0   1
## 117 154
```

Hacemos lo mismo con las variales train_crX y test_crX y X

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

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

```
summary(X)
```

```
##  TIPO_DOLOR_TORAX COLESTEROL FREC_CARDIACA_MAX ANGINA_x_EJERCICIO    OLDPEAK
##  1:166            1:288      1:  6             0:479              0      :336
##  2:195            2:444      2:107             1:333              1      : 78
##  3:451            3: 80      3:307                                2      : 67
##                              4:297                                1.5    : 49
##                              5: 94                                3      : 27
##                              6:  1                                1.2    : 23
##                                                                   (Other):232
##  PENDIENTE_ST
##  0:359
##  1:405
##  2: 48
##
##
##
##
```

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

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

```r
summary(train_X)
```

```
##  TIPO_DOLOR_TORAX COLESTEROL FREC_CARDIACA_MAX ANGINA_x_EJERCICIO    OLDPEAK
##  1:114            1:192      1:  4             0:327               0      :227
##  2:129            2:289      2: 67             1:214               1      : 49
##  3:298            3: 60      3:213                                 2      : 44
##                              4:197                                 1.5    : 35
##                              5: 60                                 3      : 18
##                              6:  0                                 1.2    : 15
##                                                                    (Other):153
##  PENDIENTE_ST
##  0:246
##  1:260
##  2: 35
##
##
##
##
```

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

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

```r
summary(test_X)
```

```
##  TIPO_DOLOR_TORAX COLESTEROL FREC_CARDIACA_MAX ANGINA_x_EJERCICIO    OLDPEAK
##  1: 52            1: 96      1:  2             0:152               0      :109
##  2: 66            2:155      2: 40             1:119               1      : 29
##  3:153            3: 20      3: 94                                 2      : 23
##                              4:100                                 1.5    : 14
##                              5: 34                                 3      :  9
##                              6:  1                                 1.2    :  8
##                                                                    (Other): 79
##  PENDIENTE_ST
##  0:113
##  1:145
##  2: 13
##
##
##
##
```

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

```r
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]      Fri Jan  6 11:45:16 2023
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 541 cases (7 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (160/7, lift 2.1)
##  TIPO_DOLOR_TORAX in {1, 2}
##  PENDIENTE_ST = 0
##  ->  class 0  [0.951]
##
## Rule 2: (193/20, lift 1.9)
##  OLDPEAK in {-1.1, -0.5, -0.1, 0, 0.2, 0.3, 0.4, 0.6, 0.7, 1.1, 1.2, 1.9,
##                      2.3, 3}
##  PENDIENTE_ST = 0
##  ->  class 0  [0.892]
##
## Rule 3: (203/29, lift 1.9)
##  TIPO_DOLOR_TORAX in {1, 2}
##  ANGINA_x_EJERCICIO = 0
##  ->  class 0  [0.854]
##
## Rule 4: (212/15, lift 1.7)
##  TIPO_DOLOR_TORAX = 3
##  PENDIENTE_ST in {1, 2}
##  ->  class 1  [0.925]
##
## Rule 5: (178/14, lift 1.7)
##  ANGINA_x_EJERCICIO = 1
##  PENDIENTE_ST in {1, 2}
##  ->  class 1  [0.917]
##
## Rule 6: (140/15, lift 1.6)
##  TIPO_DOLOR_TORAX = 3
##  OLDPEAK in {-1, -0.7, 0.1, 0.5, 0.8, 0.9, 1, 1.4, 1.5, 1.6, 1.8, 2, 2.8}
##  ->  class 1  [0.887]
##
## Default class: 1
##
##
## Evaluation on training data (541 cases):
##
```

```
##           Rules
##     ----------------
##       No      Errors
##
##        6   70(12.9%)   <<
##
##
##      (a)   (b)    <-classified as
##     ----  ----
##      223    26    (a): class 0
##       44   248    (b): class 1
##
##
##  Attribute usage:
##
##    84.47% TIPO_DOLOR_TORAX
##    84.29% PENDIENTE_ST
##    70.43% ANGINA_x_EJERCICIO
##    61.55% OLDPEAK
##
##
## Time: 0.0 secs
```

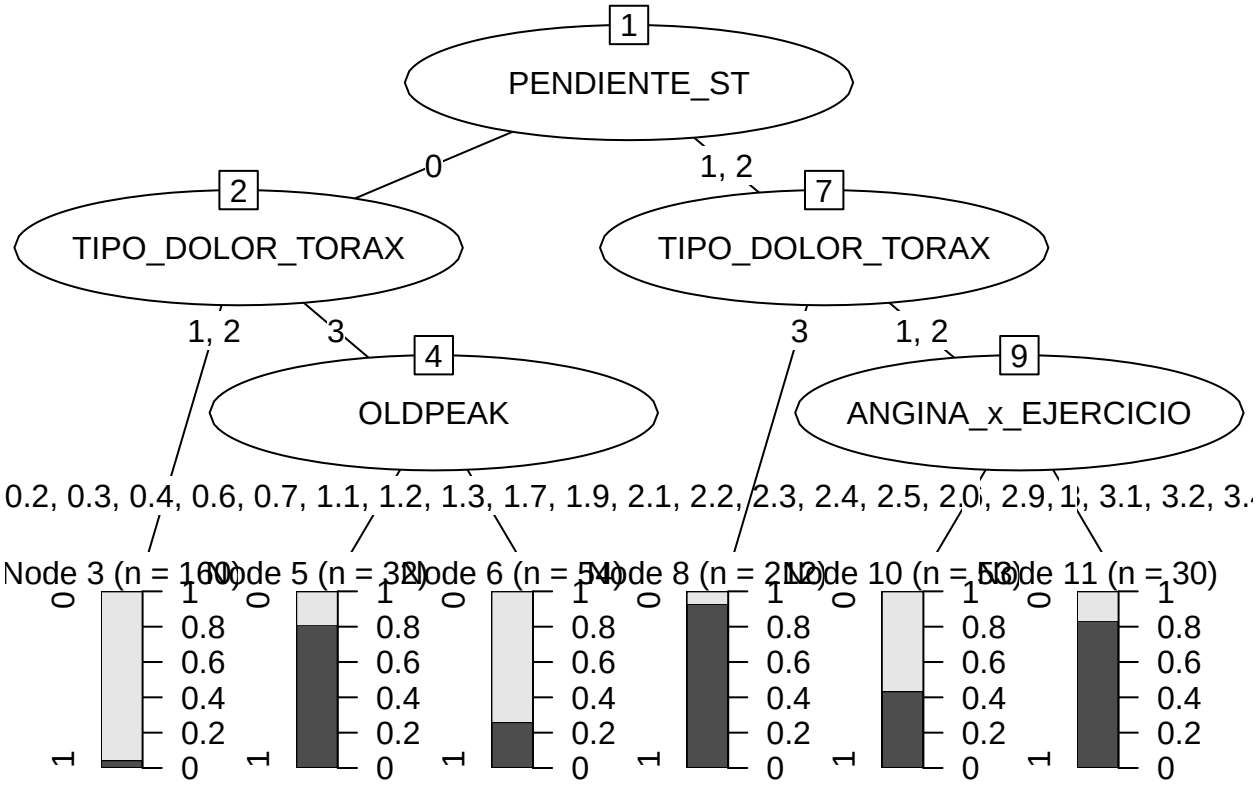
El modelo decision tree explica con dos reglas la probabilidad de sufrir una enfermedad cardiaca en función de las variables: TIPO DOLOR TORAX, COLESTEROL FREC CARDÍACA MÁX, OLDPEAK, PENDIENTE ST.

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:

```
model <- C50::C5.0(train_X, train_y)
plot(model)
```



Como podemos observar de manara visual el modelo basado en arbol de decisión, solo tiene en cuenta la variable "PENDIENTE_ST", para decidir entre si un paciente es propenso a sufrir una enfermedad cardiaca o no.

Evaluación del modelo arbol 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(predicte
```

```
## [1] "La precisión del árbol es: 82.2878 %"
```

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  94  23
##      1  25 129
```

De la matriz de confusión observamos los siguientes valores:

Verdaderos Negativos (E. CARDIACA): 94 Verdaderos Positivos (E. CARDIACA): 129 Falsos Negativos (E. CARDIACA): 25 Falsos Positivos (E. CARDIACA): 23

El modelo podria mejorarse sesgando a minimizar los falos negativos, ya que no queremos que se nos escapen del diagnostico pacientes que puedan desarrollar una enfermedad cardiaca.

Random Forest

Nos interesa saber para las predicciones que 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). As:

```
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
```

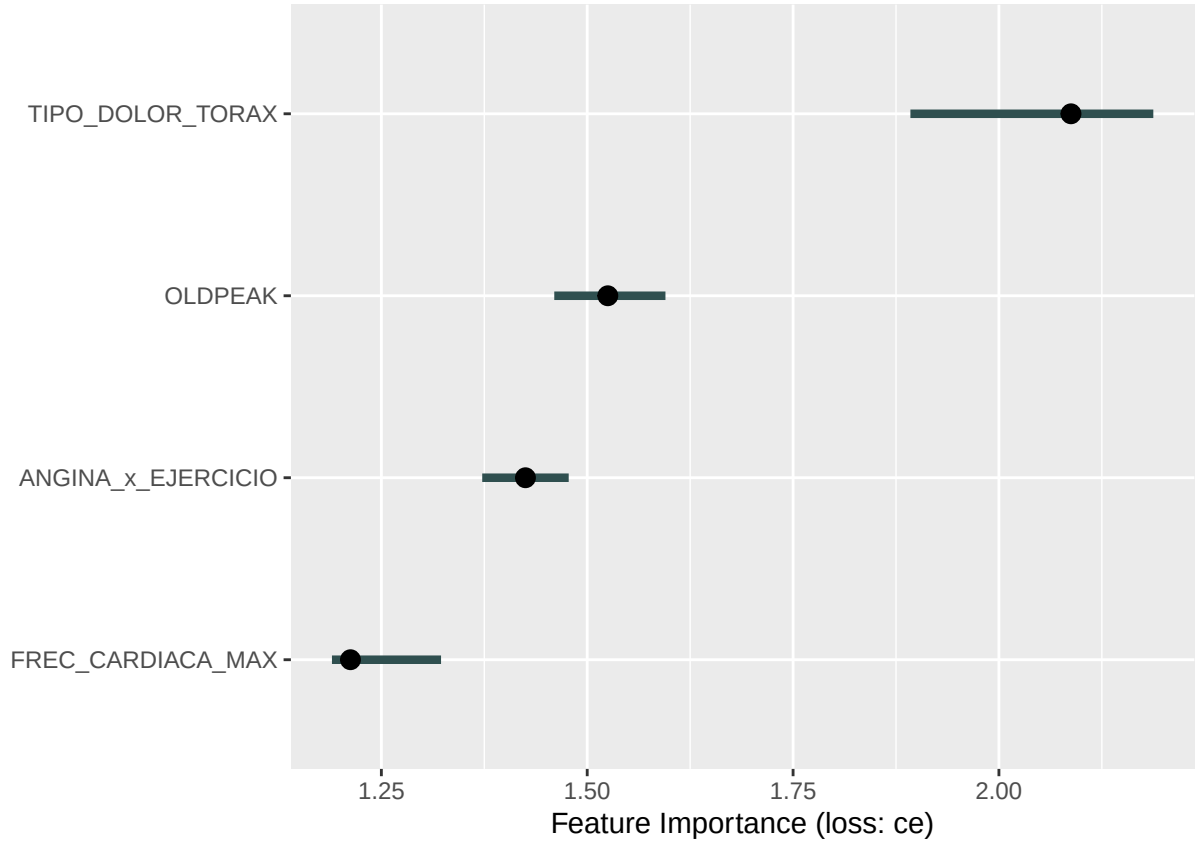
```
train_X
```

```
## # A tibble: 541 x 6
##    TIPO_DOLOR_TORAX COLESTEROL FREC_CARDIACA_MAX ANGINA_x_EJER~1 OLDPEAK PENDI~2
##    <fct>            <fct>      <fct>             <fct>           <fct>   <fct>
##  1 2                2          4                 0               2.5     1
##  2 1                2          4                 0               1.1     0
##  3 1                3          4                 0               0       0
##  4 1                2          4                 0               2       0
##  5 2                3          2                 1               0       1
##  6 3                2          4                 0               0.8     0
##  7 1                2          4                 0               0       0
##  8 3                2          4                 1               1.2     1
##  9 1                2          3                 0               0       0
## 10 3                2          3                 0               0       0
## # ... with 531 more rows, and abbreviated variable names 1: ANGINA_x_EJERCICIO,
## #   2: PENDIENTE_ST
```

```
colnames(train_X)
```

```
## [1] "TIPO_DOLOR_TORAX"   "COLESTEROL"         "FREC_CARDIACA_MAX"
## [4] "ANGINA_x_EJERCICIO" "OLDPEAK"            "PENDIENTE_ST"
```

```
train.data <- as.data.frame(cbind(train_X[c(1,3,5,4)] ,train_y))
colnames(train.data)[5] <- "E_CARDIACA"
rf <-  randomForest(E_CARDIACA ~ ., data = train.data, ntree = 50)

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

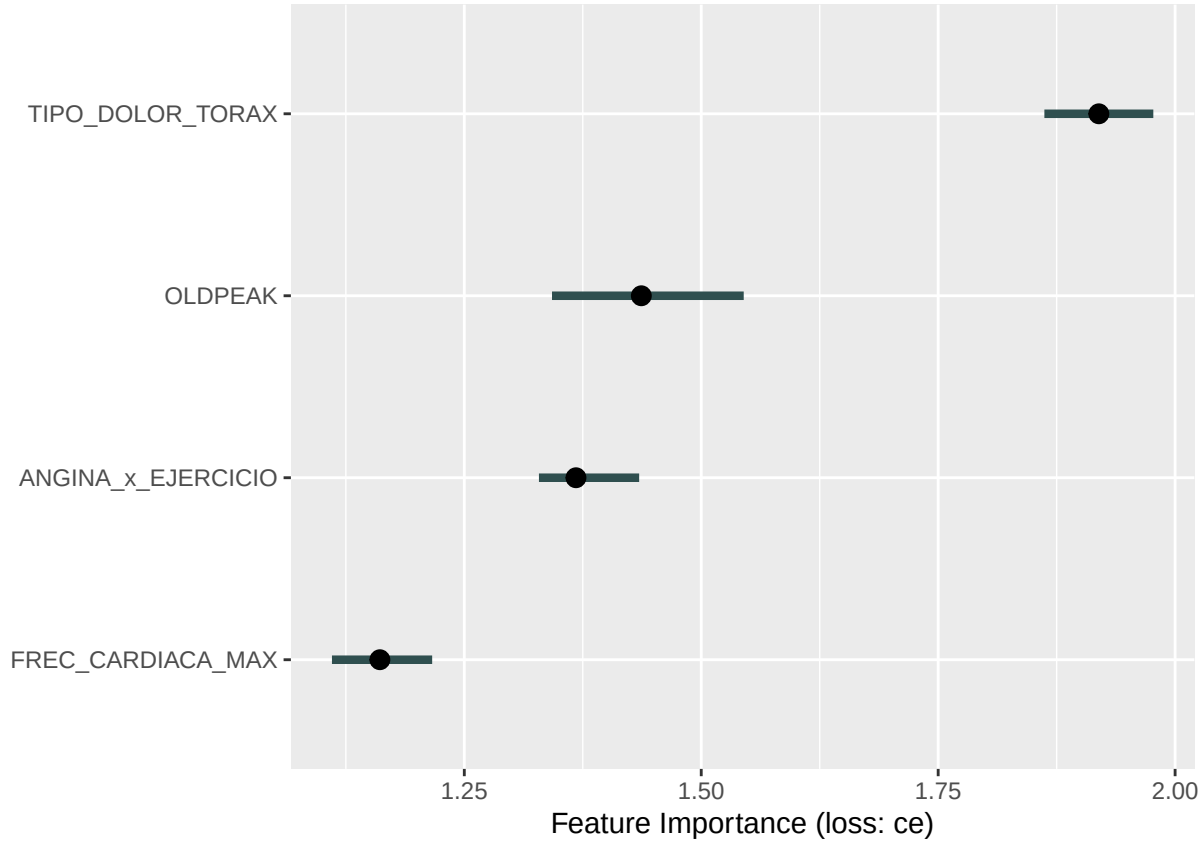



```
imp$results
```

```
##              feature importance.05 importance importance.95 permutation.error
## 1   TIPO_DOLOR_TORAX        1.8925     2.0875        2.1875         0.3086876
## 2            OLDPEAK        1.4600     1.5250        1.5950         0.2255083
## 3 ANGINA_x_EJERCICIO        1.3725     1.4250        1.4775         0.2107209
## 4  FREC_CARDIACA_MAX        1.1900     1.2125        1.3225         0.1792976
```

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




```
imp$results
```

```
##              feature importance.05 importance importance.95 permutation.error
## 1   TIPO_DOLOR_TORAX      1.862069   1.919540      1.977011         0.3086876
## 2            OLDPEAK      1.342529   1.436782      1.544828         0.2310536
## 3 ANGINA_x_EJERCICIO      1.328736   1.367816      1.434483         0.2199630
## 4  FREC_CARDIACA_MAX      1.110345   1.160920      1.216092         0.1866913
```