Universitat Oberta de Catalunya

Estudios de Informática, Multimedia y Telecomunicaciones

Minería de datos: PRA1 - Selección y preparación de un juego de datos

Autor: Eduardo Mora González

Noviembre 2021

0.1 Objetivos

El objetivo global de esta primera parte de la práctica (PRA1) consiste en seleccionar uno o varios juegos de datos, y realizar las tareas de **preparación y análisis exploratorio** con el objetivo de disponer de datos listos para después, en la segunda parte (PRA2), **aplicar algoritmos** de clustering, regresión o clasificación, demostrando la correcta asimilación de todos los aspectos trabajados durante el semestre.

---

1 Descripción de la PRA a realizar

La analítica de datos como actividad profesional, se sustenta en 3 ejes fundamentales. Uno de ellos es el profundo **conocimiento** que deberíamos tener **del problema** al que tratamos de dar respuestas mediante los estudios analíticos. Todo estudio analítico debe nacer de una necesidad por parte del **negocio** o de una necesidad de toma de decisiones basada en los datos y que resolveremos siguiendo las buenas prácticas basadas en la minería de datos. El otro aspecto importante es sin duda las **capacidades analíticas** que seamos capaces de desplegar y en este sentido, las dos prácticas de esta asignatura pretenden que el estudiante realice un recorrido sólido por este segundo eje. El tercer eje son los **datos**. Las necesidades del negocio deben concretarse con preguntas analíticas que sea posible responder a partir de los datos de que disponemos. La tarea de analizar los datos es sin duda importante, pero la tarea de identificarlos y obtenerlos va a ser para un analista un reto permanente.

Como **primera parte** del estudio analítico que nos disponemos a realizar, se pide al estudiante que complete los siguientes pasos:

1. Seleccionar un juego de datos y justificar su elección. El juego de datos deberá permitir resolver alguna pregunta analítica mediante la aplicación de algoritmos supervisados o no supervisados. El juego de datos deberá tener como mínimo 500 observaciones y debe ser distinto del usado en las PEC anteriores. El estudiante deberá visitar los siguientes portales de datos abiertos para seleccionar su juego de datos:

- **Datos abiertos**
  - Google Dataset Search (https://datasetsearch.research.google.com/)
  - Datos abiertos España (https://datos.gob.es/es/catalogo?q=&frequency=%7B%22type%22%3A+%22months%22%2C+%22value%22%3A+%221%22%7D&sort=score+desc%2C+metadata_modified+
  - Datos abiertos Madrid (https://datos.madrid.es/portal/site/egob/)
  - Datos abiertos Barcelona (https://opendata-ajuntament.barcelona.cat/es/)
  - Datos abiertos Londres (https://data.london.gov.uk/)
  - Datos abiertos New York (https://opendata.cityofnewyork.us/)
- **Conjuntos de datos para aprendizaje automático e investigación**
  - UCI Machine Learning (https://archive.ics.uci.edu/ml/datasets.php)
  - Datasets for machine-learning research (Wikipedia) (https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research)

- Kaggle datasets (https://www.kaggle.com/datasets)

Algún ejemplo del tipo de datos y problemas que podrían elegirse:

Aprobación de tarjetas de crédito (https://www.kaggle.com/rikdifos/credit-card-approval-prediction) Una tarea de clasificación binaria para predecir si las personas pueden presentar un riesgo de incumplimiento de préstamos de tarjetas de crédito.

Ventas de comercio electrónico (https://www.kaggle.com/carrie1/ecommerce-data): Predicción de ventas y transacciones en una tienda online. Un problema clásico de predicción de series temporales.

Debéis tener en cuenta que deberéis usar este mismo conjunto de datos, una vez procesado y acondicionado, en los ejercicios de modelado dedatos de la segunda parte de la práctica (PRA2).

2. Plantear un problema de analítica de datos sobre el conjunto de datos seleccionado, detallando los objetivos analíticos y desarrollando una metodología para resolverlos de acuerdo con lo que se ha practicado en las PEC y lo que se ha aprendido en el material didáctico.
3. Realizar un análisis exploratorio del juego de datos seleccionado, utilizando las visualizaciones que crea necesrias para ilustrar su análisis.
4. Realizar las tareas de limpieza y acondicionado necesarias para que los datos puedan ser usados en los consiguientes procesos de modelado.
5. Analizar las diferentes variables y realizar las transformaciones necesarias para la optimización de los procesos de modelado.
6. Realizar un análisis de componentes principales (PCA) o descomposición de valores singulares (SVD) sobre el juego de datos explicando los aspectos más destacados del análisis. Utilizar las componentes obtenidas para visualizar el conjunto de datos, y en su caso, poeéis usar las componentes como variables en el proceso de modelado. Se valorará si además profundizais en alguna otra técnica explicada en el módulo docente de **Preprocesado y gestión de características**.

1.1 Recursos

1.1.1 Recursos Básicos

Material docente proporcionado por la UOC.

1.1.2 Recursos de programación

- Incluimos en este apartado una lista de recursos de programación para minería de datos donde podréis encontrar ejemplos, ideas e inspiración:
  - Material adicional del libro: Minería de datos Modelos y Algoritmos (http://oer.uoc.edu/libroMD/)
  - Espacio de recursos UOC para ciencia de datos (http://datascience.recursos.uoc.edu/es/)
  - Buscador de código R (https://rseek.org/)
  - Colección de cheatsheets en R (https://rstudio.com/resources/cheatsheets/)

1.2 Criterios de valoración

Para todas las PRA es **necesario documentar** en cada apartado del ejercicio práctico que se ha hecho y como se ha hecho. Asimismo, todas las decisiones y conclusiones deberán ser presentados de forma razonada y clara, especificando todos y cada uno de los pasos que se hayan llevado a cabo para su resolución.

1.3 Formato y fecha de entrega PRA1

El formato de entrega es: **usernameestudiant-PRA1.Rmd** y **usernameestudiant-PRA1.html** (o .pdf/.docx)
Se debe entregar la PRA1 en el buzón de entregas del aula

1.4 Nota: Propiedad intelectual

> A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por lo tanto comprensible hacerlo en el marco de una práctica de los estudios de Informática, Multimedia y Telecomunicación de la UOC, siempre y cuando esto se documente claramente y no suponga plagio en la práctica.

> Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se debe presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra esta protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL ...). El estudiante deberá asegurarse de que la licencia no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente tendrá que asumir que la obra esta protegida por copyright.

> Deberéis, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.

2 Desarrollo de la práctica

Instalamos y cargamos las librerías necesarias.

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
if (!require('GGally')) install.packages('GGally'); library(GGally)
if (!require('DataExplorer')) install.packages("DataExplorer"); library(DataExplorer)
if (!require('dlookr')) install.packages("dlookr"); library(dlookr)
if (!require('tidymodels')) install.packages("tidymodels"); library(tidymodels)
if (!require('flextable')) install.packages("flextable"); library(flextable)
if (!require('corrplot')) install.packages("corrplot"); library(corrplot)
if (!require('textshape')) install.packages("textshape"); library(textshape)
if (!require('stats')) install.packages("stats"); library(stats)
if (!require('FactoMineR')) install.packages("FactoMineR"); library(FactoMineR)
if (!require('factoextra')) install.packages("factoextra"); library(factoextra)
```

2.1 Elección del conjunto de datos

En Europa, el paro cardiaco es una de las primeras causas de mortalidad y en España fallecen en torno a 100 personas al día por este suceso (https://fundaciondelcorazon.com/prensa/notas-de-prensa/2900-solo-el-30-de-espanoles-sabe-realizar-la-reanimacion-cardio-pulmonar-rcp-.html (https://fundaciondelcorazon.com/prensa/notas-de-prensa/2900-solo-el-30-de-espanoles-sabe-realizar-la-reanimacion-cardio-pulmonar-rcp-.html)), esto representa aproximadamente el 31% de las muertes a nivel mundial.

Por esta razón, se han seleccionado dos conjuntos de datos, el primer conjunto (https://www.kaggle.com/fedesoriano/heart-failure-prediction?select=heart.csv (https://www.kaggle.com/fedesoriano/heart-failure-prediction?select=heart.csv)) contiene 12 características y el segundo (https://www.kaggle.com/ronitf/heart-disease-uci (https://www.kaggle.com/ronitf/heart-disease-uci)) contiene 13 características. Aunque el número de características que contienen son distintas, muchas son comunes entre los dos y esto permitirá crear un conjunto de datos más completo.

Los dos conjuntos de datos han sido elegidos por las características que estos contienen, ya que son los parámetros típicos usados en los estudios de problemas del corazón, y es por eso por lo que tras el análisis de estos se puede sacar unas conclusiones bastantes interesantes.

Finalmente, se puede decir que el objetivo buscado es predecir la posibilidad de que una persona tenga un alto riesgo de ser diagnosticado como un paciente cardíaco a través de las diversas características. Para llegar a al objetivo se tiene pensado realizar diversos métodos de análisis para así relacionar las diversas características para obtener unos parámetros finales y así concluir la posibilidad de que una persona tenga o no una enfermedad cardiaca.

2.2 Exploración del conjunto de datos

A continuación, se van a exponer las diferentes características de los conjuntos de datos.

2.2.1 Características del Primer conjunto de datos

Del primer conjunto, como se ha mencionado anteriormente tenemos 12 características distintas:

- **Age:** edad del paciente [años]
- **Sex:** sexo del paciente [M: Masculino, F: Femenino]
- **ChestPainType:** tipo de dolor de pecho [TA: angina típica, ATA: angina atípica, NAP: dolor no anginal, ASY: asintomático]
- **RestingBP:** presión arterial en reposo [mm Hg]
- **Cholesterol:** colesterol sérico [mm / dl]
- **FastingBS:** azúcar en sangre en ayunas [1: si BS en ayunas> 120 mg / dl, 0: en caso contrario]
- **RestingECG:** resultados del electrocardiograma en reposo [Normal: Normal, ST: con anomalía de la onda ST-T (inversiones de la onda T y / o elevación o depresión del ST> 0,05 mV), LVH: que muestra una hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes]
- **MaxHR:** frecuencia cardíaca máxima alcanzada [Valor numérico entre 60 y 202]
- **ExerciseAngina:** angina inducida por el ejercicio [Y: Sí, N: No]
- **Oldpeak:** oldpeak = ST [Valor numérico medido en depresión]
- **ST_Slope:** la pendiente del segmento ST del ejercicio pico [Up: uploping, Flat: flat, Down: downsloping]
- **HeartDisease:** clase de salida [1: enfermedad cardíaca, 0: Normal]

2.2.2 Características del Segundo conjunto de datos

El segundo conjunto de datos tiene las siguientes características:

- **Age:** la edad de la persona en años
- **sex:** el sexo de la persona [1 = hombre, 0 = mujer]
- **cp:** el dolor torácico experimentado [valor 0: angina típica, valor 1: angina atípica, valor 2: dolor no anginoso, valor 3: asintomático]
- **trestbps:** la presión arterial en reposo de la persona [mm Hg al ingreso en el hospital]
- **chol:** la medición del colesterol de la persona en mg / dl
- **fbs:** nivel de azúcar en sangre en ayunas de la persona [> 120 mg / dl, 1 = verdadero; 0 = falso]
- **restecg:** medición electrocardiográfica en reposo [0 = normal, 1 = con anomalía de la onda ST-T, 2 = mostrando hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes]
- **thalach:** frecuencia cardíaca máxima alcanzada por la persona
- **exang:** angina inducida por ejercicio [1 = sí; 0 = no]
- **oldpeak:** depresión del ST inducida por el ejercicio en relación con el reposo.
- **Slope:** la pendiente del segmento ST de ejercicio pico [Valor 0: pendiente ascendente, Valor 1: plano, Valor 2: pendiente descendente]
- **ca:** Número de vasos principales (0-3) coloreados por la floración
- **Thal:** Trastorno de la sangre llamado talasemia [3 = normal; 6 = defecto fijo; 7 = defecto reversible]
- **target:** clase de salida [1: enfermedad cardíaca, 0: Normal]

2.2.3 Características comunes y no comunes de los dos juego de datos

Como se puede observar, las características de los dos conjuntos de datos que coinciden son:

Comparación de características

Primer conjunto de datos	Segundo conjunto de datos	Significado
Age	Age	Edad de la persona
Sex	Sex	Sexo de la persona
ChestPainType	cp	Tipo dolor torácico
RestingBP	trestbps	Presión arterial en reposo
Cholesterol	chol	colesterol de la persona
FastingBS	fbs	Nivel de azúcar en sangre
RestingECG	restecg	ECG en reposo
MaxHR	thalach	Frecuencia cardíaca máxima
ExerciseAngina	exang	Angina inducida por ejercicio
Oldpeak	oldpeak	depresión del ST
ST_Slope	Slope	pendiente del segmento ST
HeartDisease	target	¿Enfermedad Cardiaca?

Las únicas características que no se encuentran en el primer conjunto de datos son:

- **ca:** Número de vasos principales coloreados por la floración
- **Thal:** Trastorno de la sangre llamado talasemia

2.2.4 Carga de los conjuntos de datos

Una vez identificadas las características, cargamos los archivos para un análisis exploratorio del conjunto de datos.

```
#Cargamos el primer fichero
datos1 <- read.csv('heart.csv')

#Cargamos el segundo fichero
datos2 <- read.csv('heart_1.csv')

#Filas del primer fichero
filas_1 = dim(datos1)[1]

#Filas del segundo fichero
filas_2 = dim(datos2)[1]
```

Ahora vamos a ver las estructura de los juegos de datos

```
#Verificamos la estructura del primer juego
str(datos1)
```

```
## 'data.frame':    918 obs. of  12 variables:
##  $ Age           : int  40 49 37 48 54 39 45 54 37 48 ...
##  $ Sex           : chr  "M" "F" "M" "F" ...
##  $ ChestPainType : chr  "ATA" "NAP" "ATA" "ASY" ...
##  $ RestingBP     : int  140 160 130 138 150 120 130 110 140 120 ...
##  $ Cholesterol   : int  289 180 283 214 195 339 237 208 207 284 ...
##  $ FastingBS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RestingECG    : chr  "Normal" "Normal" "ST" "Normal" ...
##  $ MaxHR         : int  172 156 98 108 122 170 170 142 130 120 ...
##  $ ExerciseAngina: chr  "N" "N" "N" "Y" ...
##  $ Oldpeak       : num  0 1 0 1.5 0 0 0 0 1.5 0 ...
##  $ ST_Slope      : chr  "Up" "Flat" "Up" "Flat" ...
##  $ HeartDisease  : int  0 1 0 1 0 0 0 0 1 0 ...
```

```
#Verificamos la estructura del segundo juego
str(datos2)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ ï..age  : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Vamos ahora a sacar estadísticas básicas de los juegos de datos

```
#Estadísticas básica del primer juego
summary(datos1)
```

```
##       Age            Sex            ChestPainType        RestingBP      Cholesterol      FastingBS       Resting
ECG            MaxHR       ExerciseAngina
##  Min.   :28.00   Length:918         Length:918         Min.   :  0.0   Min.   :  0.0   Min.   :0.0000   Length:9
18         Min.   : 60.0   Length:918
##  1st Qu.:47.00   Class :character   Class :character   1st Qu.:120.0   1st Qu.:173.2   1st Qu.:0.0000   Class :c
haracter   1st Qu.:120.0   Class :character
##  Median :54.00   Mode  :character   Mode  :character   Median :130.0   Median :223.0   Median :0.0000   Mode  :c
haracter   Median :138.0   Mode  :character
##   Mean    :53.51                                                      Mean    :132.4    Mean    :198.8    Mean    :0.2331
Mean   :136.8
##   3rd Qu.:60.00                                                       3rd Qu.:140.0    3rd Qu.:267.0    3rd Qu.:0.0000
3rd Qu.:156.0
##   Max.    :77.00                                                     Max.    :200.0    Max.    :603.0    Max.    :1.0000
Max.   :202.0
##     Oldpeak          ST_Slope          HeartDisease
##  Min.   :-2.6000   Length:918         Min.   :0.0000
##  1st Qu.: 0.0000   Class :character   1st Qu.:0.0000
##  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
```

```
#Estadísticas básica del segundo juego
summary(datos2)
```

```
##      ï..age           sex               cp           trestbps          chol            fbs            restecg
thalach          exang           oldpeak
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000   Min.   :0.000
0   Min.   : 71.0   Min.   :0.0000   Min.   :0.00
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.000
0   1st Qu.:133.5   1st Qu.:0.0000   1st Qu.:0.00
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0   Median :240.0   Median :0.0000   Median :1.000
0   Median :153.0   Median :0.0000   Median :0.80
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6   Mean   :246.3   Mean   :0.1485   Mean   :0.528
1   Mean   :149.6   Mean   :0.3267   Mean   :1.04
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0   3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.000
0   3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:1.60
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0   Max.   :564.0   Max.   :1.0000   Max.   :2.000
0   Max.   :202.0   Max.   :1.0000   Max.   :6.20
##      slope            ca              thal           target
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000
##  Median :1.000   Median :0.0000   Median :2.000   Median :1.0000
##  Mean   :1.399   Mean   :0.7294   Mean   :2.314   Mean   :0.5446
##  3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :2.000   Max.   :4.0000   Max.   :3.000   Max.   :1.0000
```

2.3 Preprocesado y gestión de características

2.3.1 Valores nulos del conjunto de los datos

Estadísticas de valores vacíos del primer juego de datos

```
colSums(is.na(datos1))
```

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

```
colSums(datos1=="")
```

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

Estadísticas de valores vacíos del segundo juego de datos

```
colSums(is.na(datos2))
```

```
##   ï..age       sex        cp  trestbps      chol       fbs   restecg   thalach     exang   oldpeak     slope        ca
thal   target
##        0         0         0         0         0         0         0         0         0         0         0         0
0        0
```

```
colSums(datos2=="")
```

```
##   ï..age       sex        cp  trestbps      chol       fbs   restecg   thalach     exang   oldpeak     slope        ca
thal   target
##        0         0         0         0         0         0         0         0         0         0         0         0
0        0
```

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

2.3.2 Normalización del conjunto de los datos

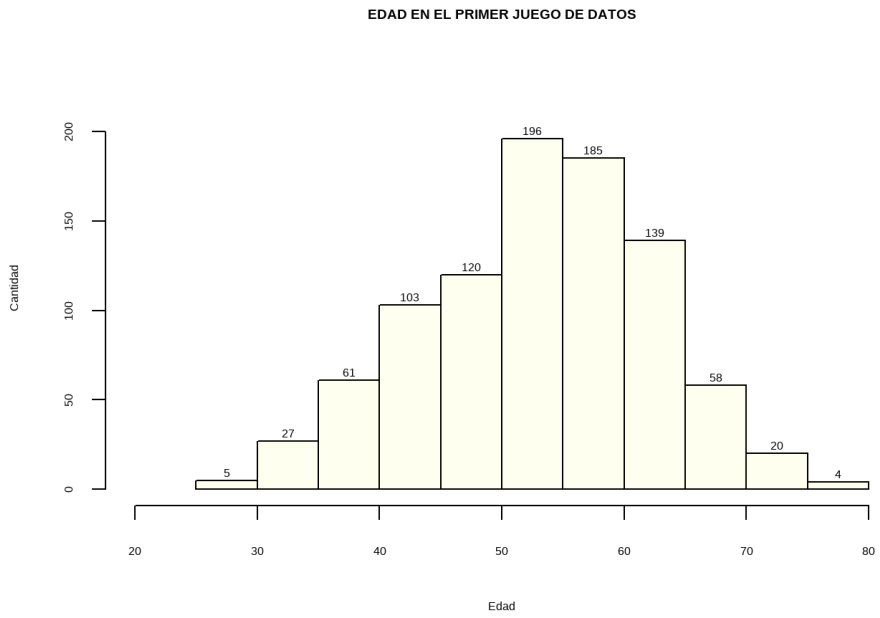
Ahora que hemos comprobado que no tenemos valores nulos, se va a proceder a la normalización de los dos conjuntos de datos. La importancia de este proceso es para que a la hora de juntar los dos juegos de datos estén todos en la misma escala de valores y que así se pueda hacer un merge limpio y rápido.

Para la normalización, se hará un análisis de las características comunes comparando una por una de cada uno de los conjuntos de datos. Además, una vez normalizado se analizarán las características para ver posibles valores incorrectos y poder corregirlos.

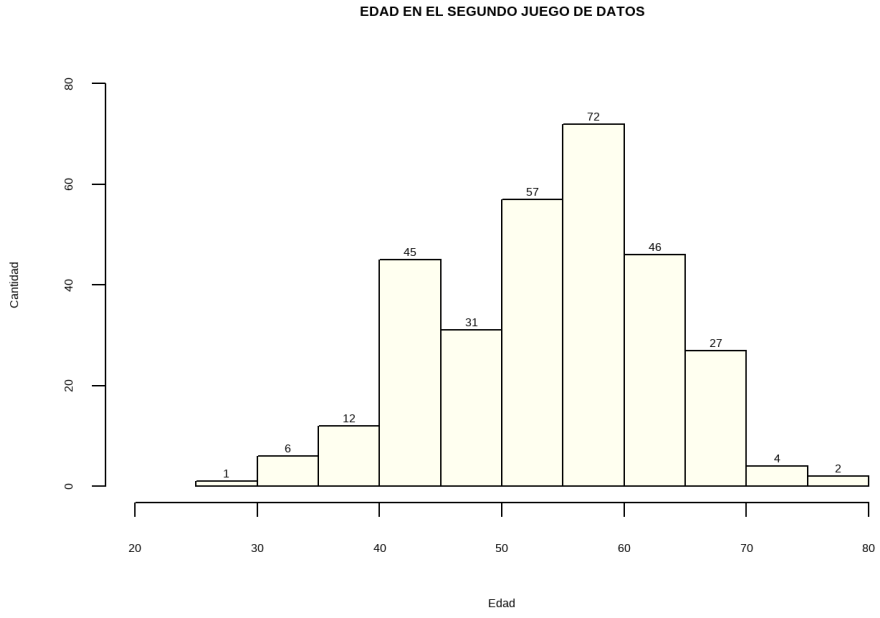
- **EDAD**

Como se puede comprobar en las estadísticas del primer conjunto de datos las edades van desde los 28 hasta los 77 años, mientras que en el segundo van desde los 29 hasta los 77 años.

```
#Histograma de la característica edad del primer conjunto de datos
h1 <- hist(datos1$Age, xlab="Edad", col="ivory", ylab="Cantidad", main="EDAD EN EL PRIMER JUEGO DE DATOS", ylim = c
(0, 225), xlim = c(20,80))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```




```
#Histograma de la característica edad del segundo conjunto de datos
h2 <- hist(datos2$ï..age, xlab="Edad", col="ivory", ylab="Cantidad", main="EDAD EN EL SEGUNDO JUEGO DE DATOS", ylim
= c(0, 80), xlim = c(20,80))
text(h2$mids,h2$counts,labels=h2$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.

Una diferencia bastante clara es en la franja de entre los 40 y 45 años, que en el primer conjunto de datos hay un crecimiento de los datos de manera progresiva, mientras que en el segundo existen un crecimiento notable de los datos bastante peculiar en ese rango.

- **SEXO**

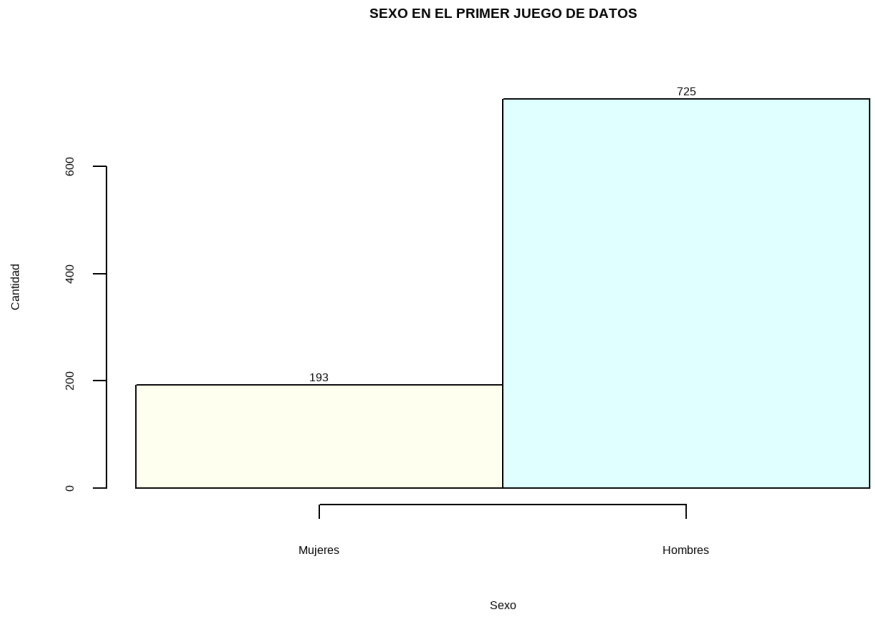
En esta característica observamos que en primer conjunto de datos están identificado con las variables M (hombre) y F (mujer) mientras que en el segundo juego de datos tenemos 1 (hombre) y 0 (mujer).

Entonces se va a normalizar el primer conjunto de datos para que sea como el segundo, vamos a definir el valor 1 para el hombre y el valor 0 para la mujer.

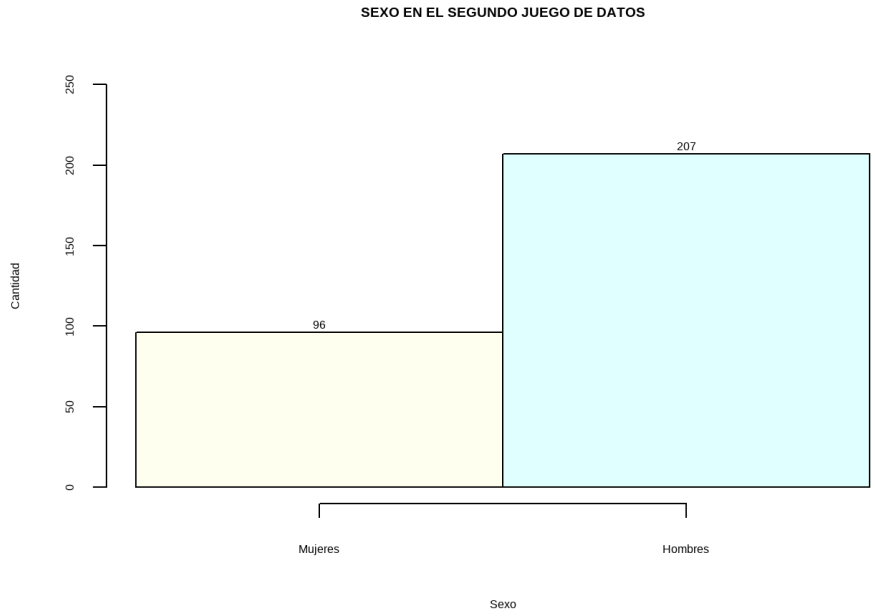
```
#Cambiamos las letras por los números
datos1$Sex [datos1$Sex == "M"] <- 1
datos1$Sex [datos1$Sex == "F"] <- 0

#Pasamos de carácter a numérico
datos1$Sex <- as.numeric(datos1$Sex)
```

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

```
#Histograma de la característica sexo del primer conjunto de datos
h1 <- hist(datos1$Sex, xlab="Sexo", col=c("ivory", "lightcyan"), ylab="Cantidad", main="SEXO EN EL PRIMER JUEGO DE
 DATOS", breaks = 2, ylim = c(0, 750), axes = FALSE)
text(h1$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)
```




```
#Histograma de la característica sexo del segundo conjunto de datos
h2 <-hist(datos2$sex, xlab="Sexo", col=c("ivory", "lightcyan"), ylab="Cantidad", main="SEXO EN EL SEGUNDO JUEGO DE
 DATOS", breaks = 2, ylim = c(0, 250), axes = FALSE)
text(h2$mids,h2$counts,labels=h2$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75), cex.axis=1, labels = c("Mujeres","Hombres" ))
axis(2)
```



Tras la normalización y la exploración de los datos, nos damos cuenta de que existen mas registros de hombres que de mujeres en los dos conjuntos de datos.

- **TIPO DE DOLOR TORÁCICO (ChestPainType, cp)**

Nos damos cuenta de que el primer conjunto de datos viene identificado por 4 variables categóricas (TA: angina típica, ATA: angina atípica, NAP: dolor no anginal, ASY: asintomático) mientras en el segundo conjunto de datos por valores numérico y cada valor asignado a una causa (valor 0: angina típica, valor 1: angina atípica, valor 2: dolor no anginoso, valor 3: asintomático).

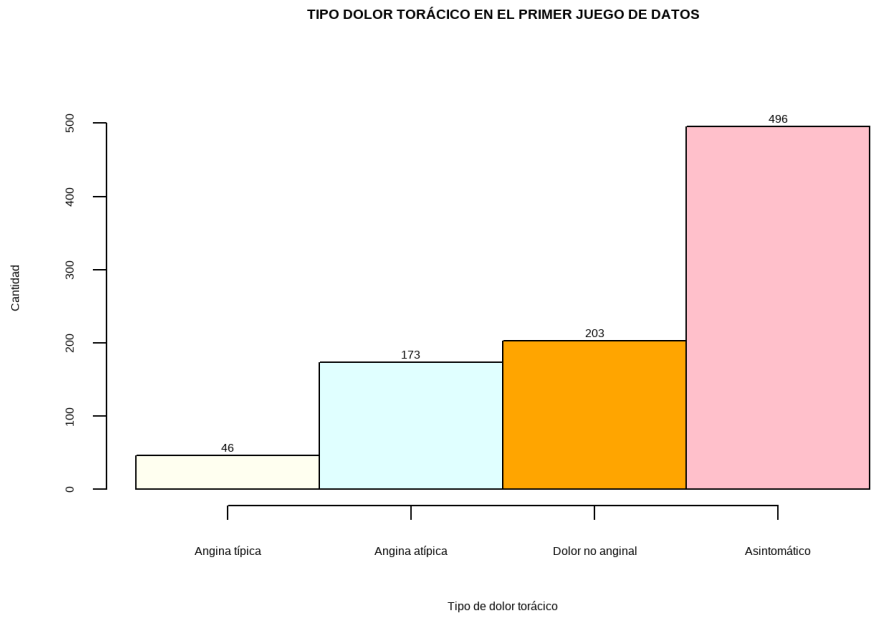
La normalización se hará para el primer conjunto de datos, asignando los valores (que son los del segundo conjunto de datos) de la siguiente manera:

```
+ 0 = TA
+ 1 = ATA
+ 2 = NAP
+ 3 = ASY
```

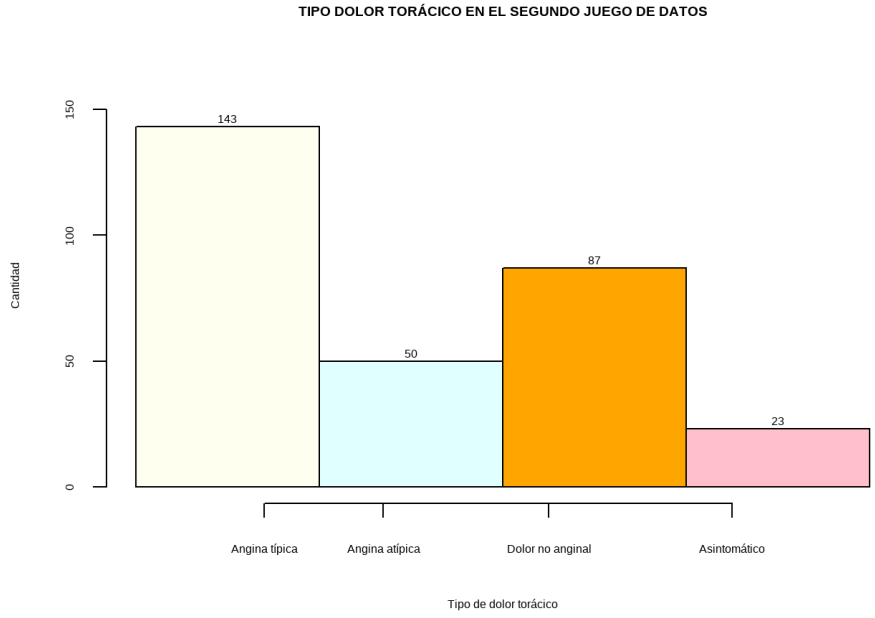
```
#Cambiamos las letras por los números
datos1$ChestPainType [datos1$ChestPainType == "TA"]  <- 0
datos1$ChestPainType [datos1$ChestPainType == "ATA"] <- 1
datos1$ChestPainType [datos1$ChestPainType == "NAP"] <- 2
datos1$ChestPainType [datos1$ChestPainType == "ASY"] <- 3

#Pasamos de carácter a numérico
datos1$ChestPainType <- as.numeric(datos1$ChestPainType)
```

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

```
#Histograma de la característica Tipo de dolor torácico del primer conjunto de datos
h1 <- hist(datos1$ChestPainType, xlab="Tipo de dolor torácico", col= c("ivory", "lightcyan", "ORANGE", "PINK"), yla
b="Cantidad", main="TIPO DOLOR TORÁCICO EN EL PRIMER JUEGO DE DATOS", ylim = c(0, 550),axes = FALSE, breaks=seq(min
(datos1$ChestPainType)-0.5, max(datos1$ChestPainType)+0.5, by=1) )
text(h1$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)
```




```
#Histograma de la característica Tipo de dolor torácico del segundo conjunto de datos
h2 <- hist(datos2$cp, xlab="Tipo de dolor torácico", col= c("ivory", "lightcyan", "ORANGE", "PINK"), ylab="Cantida
d", main="TIPO DOLOR TORÁCICO EN EL SEGUNDO JUEGO DE DATOS", ylim = c(0, 160),axes = FALSE, breaks=seq(min(datos2$c
p)-0.5, max(datos2$cp)+0.5, by=1) )
text(h2$mids,h2$counts,labels=h2$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.2,0.85,1.75,2.75), cex.axis=1, labels = c("Angina típica", "Angina atípica","Dolor no anginal", "As
intomático" ))
axis(2)
```



Como se puede comprobar, en los dos conjuntos de datos tenemos diversas proporciones del tipo de dolor torácico, lo que supone tener mas variedad a la hora de poder sacar conclusiones.

- **PRESIÓN ARTERIAL EN REPOSO (RestingBP y trestbps)**

Como se muestran en las estadísticas esta característica son de tipo numérico y en el primer conjunto de datos va desde 0 hasta 200 y en el segundo de 94 a 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(datos1$RestingBP[datos1$RestingBP == 0])
```

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

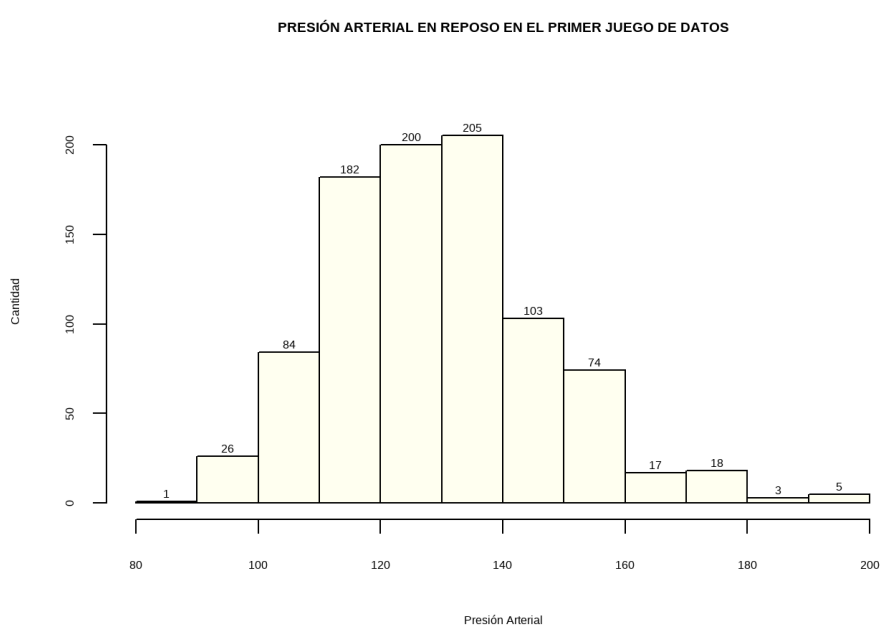
Como solo aparece una vez, se le asignará un valor por defecto. El valor por defecto será el más común.

```
#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(datos1$RestingBP)

#Asignamos el valor
datos1$RestingBP[datos1$RestingBP == 0] <- BP_comun

#vemos las estaditicas del dato
summary(datos1$RestingBP)
```

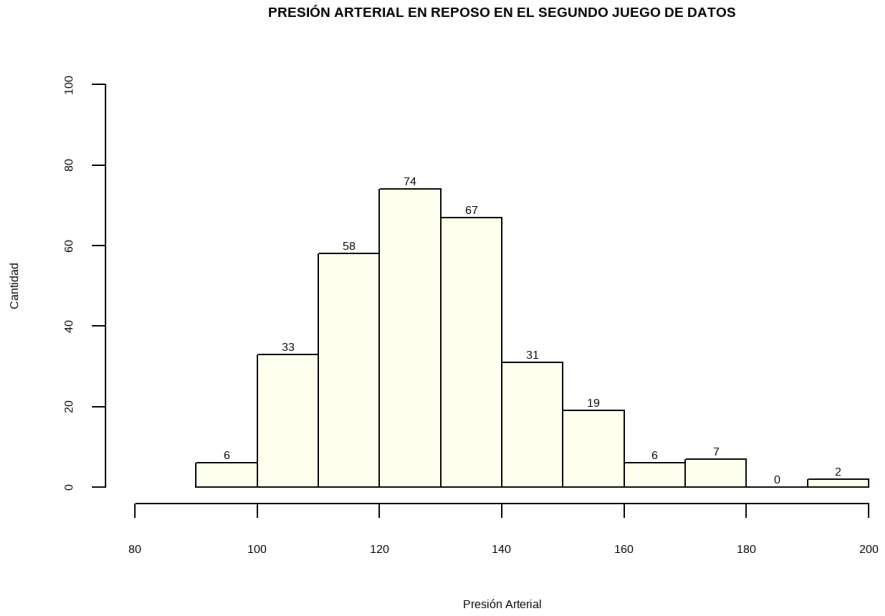
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    80.0   120.0   130.0   132.5   140.0   200.0
```

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(datos1$RestingBP, xlab="Presión Arterial", col="ivory", ylab="Cantidad", main="PRESIÓN ARTERIAL EN REPOS
O EN EL PRIMER JUEGO DE DATOS", ylim = c(0, 225), xlim = c(80,200))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```




```
#Histograma de la característica Presión Arterial del segundo conjunto de datos
h1 <- hist(datos2$trestbps, xlab="Presión Arterial", col="ivory", ylab="Cantidad", main="PRESIÓN ARTERIAL EN REPOSO
EN EL SEGUNDO JUEGO DE DATOS", ylim = c(0, 100), xlim = c(80,200))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```



Se puede observar que el grueso de los datos está entre 100 y 160 en los dos conjuntos de datos.

- **COLESTEROL (Cholesterol y chol)**

La siguiente característica en ambos conjuntos de datos es de tipo numérico. Al igual que en la presión arterial en reposo, en el primer data set tenemos valores 0 que debemos analizar, mientras que en el segundo data set tenemos datos que abarcan desde el 126 hasta 564.

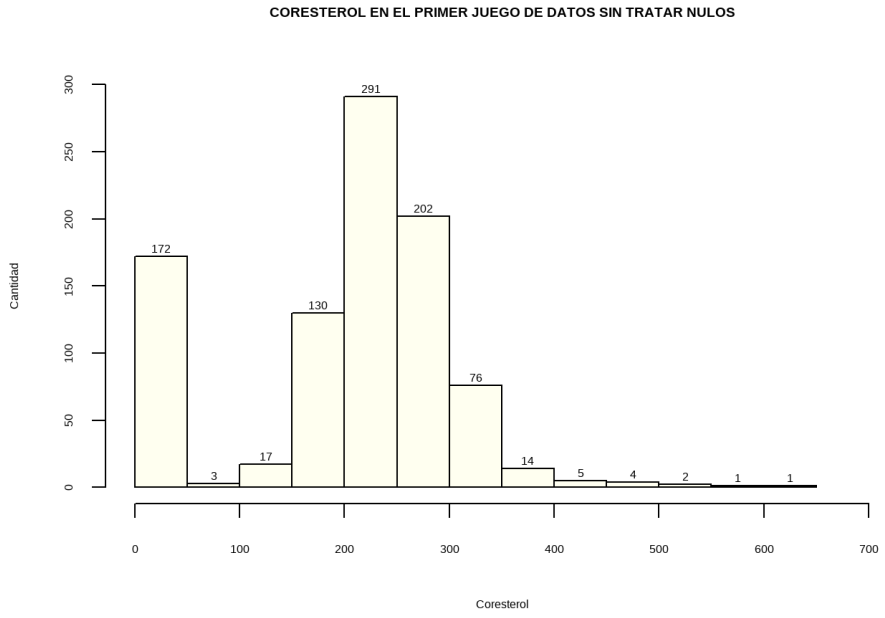
Lo primero que se va a hacer es obtener el numero de casos que el coresterol es 0, y se consideraran las diversas formas de tratar estos datos.

```
#Veces que aparece el valor cero en la presion arterial
length(datos1$RestingBP[datos1$Cholesterol == 0])
```

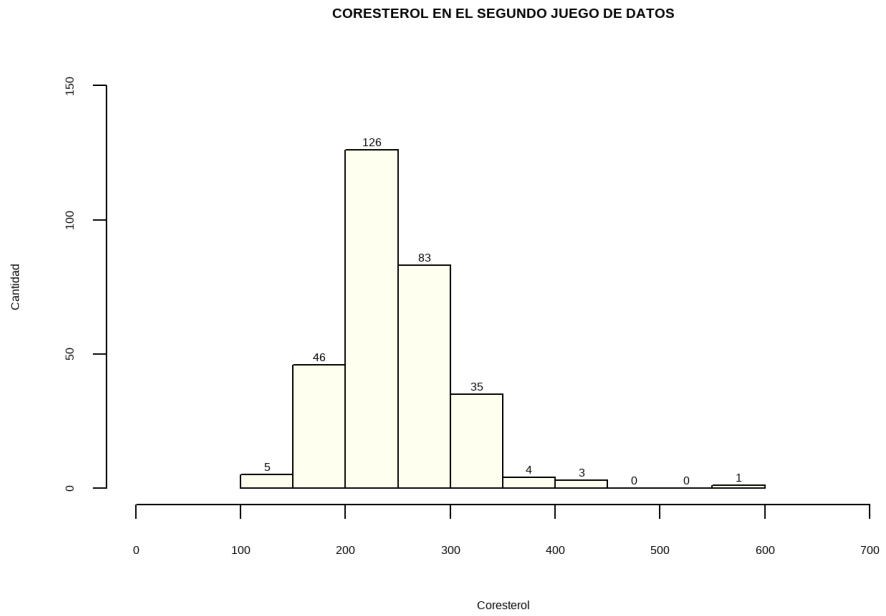
```
## [1] 172
```

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.

```
#Histograma de la característica Coresterol del primer conjunto de datos
h1 <- hist(datos1$Cholesterol, xlab="Coresterol", col="ivory", ylab="Cantidad", main="CORESTEROL EN EL PRIMER JUEGO
DE DATOS SIN TRATAR NULOS", ylim = c(0,300), xlim = c(0, 700))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```



```
#Histograma de la característica Coresterol del segundo conjunto de datos
h1 <- hist(datos2$chol, xlab="Coresterol", col="ivory", ylab="Cantidad", main="CORESTEROL EN EL SEGUNDO JUEGO DE DA
TOS", ylim = c(0,150), xlim = c(0, 700))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```



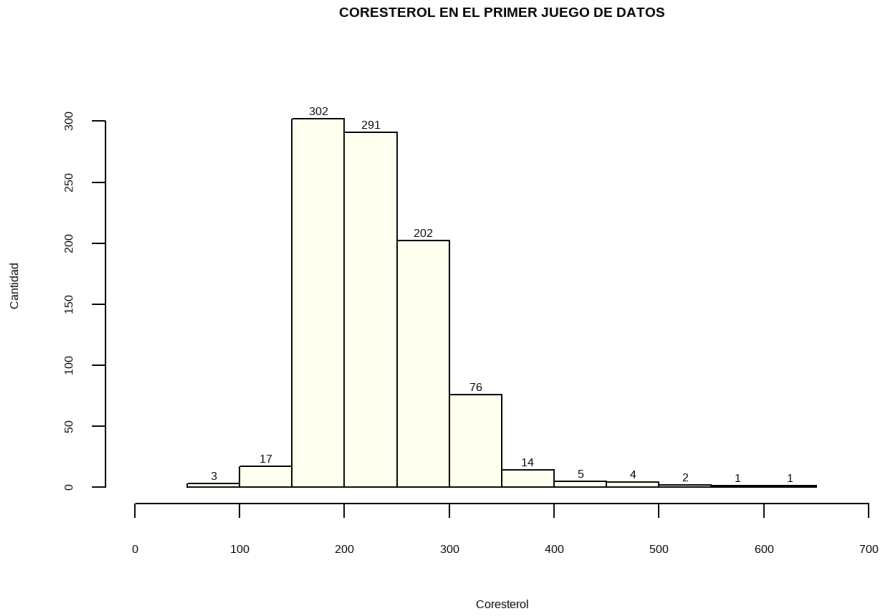
Tras analizar la gráfica y para no perder estos datos, se le asignaran un valor por defecto, que será la media de los datos. Esta decisión se ha tomado ya que poner el más común, nos crearía un conjunto de datos muy distintos entre unas medidas y otras, mientras que poner la media sería un valor que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

```
#Calculamos el valor más comun
coresterol_media <- mean(datos1$Cholesterol)

#Asignamos el valor truncado para evitar decimales
datos1$Cholesterol[datos1$Cholesterol == 0] <- trunc(coresterol_media)

#vemos las estaditicas del dato
summary(datos1$RestingBP)
```

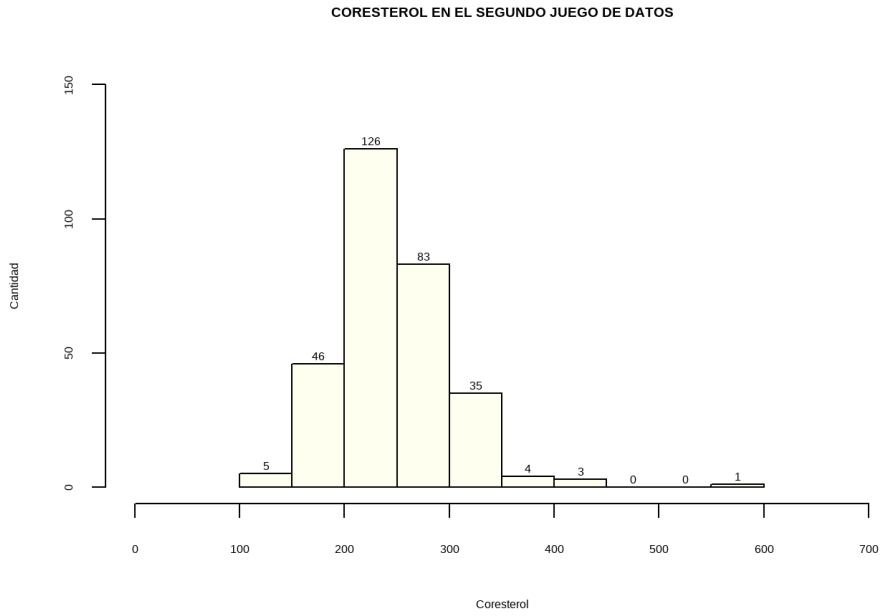
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    80.0   120.0   130.0   132.5   140.0   200.0
```

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

```
#Histograma de la característica Coresterol del primer conjunto de datos
h1 <- hist(datos1$Cholesterol, xlab="Coresterol", col="ivory", ylab="Cantidad", main="CORESTEROL EN EL PRIMER JUEGO
DE DATOS", ylim = c(0,330), xlim = c(0, 700))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```




```
#Histograma de la característica Coresterol del segundo conjunto de datos
h1 <- hist(datos2$chol, xlab="Coresterol", col="ivory", ylab="Cantidad", main="CORESTEROL EN EL SEGUNDO JUEGO DE DA
TOS", ylim = c(0,150), xlim = c(0, 700))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```

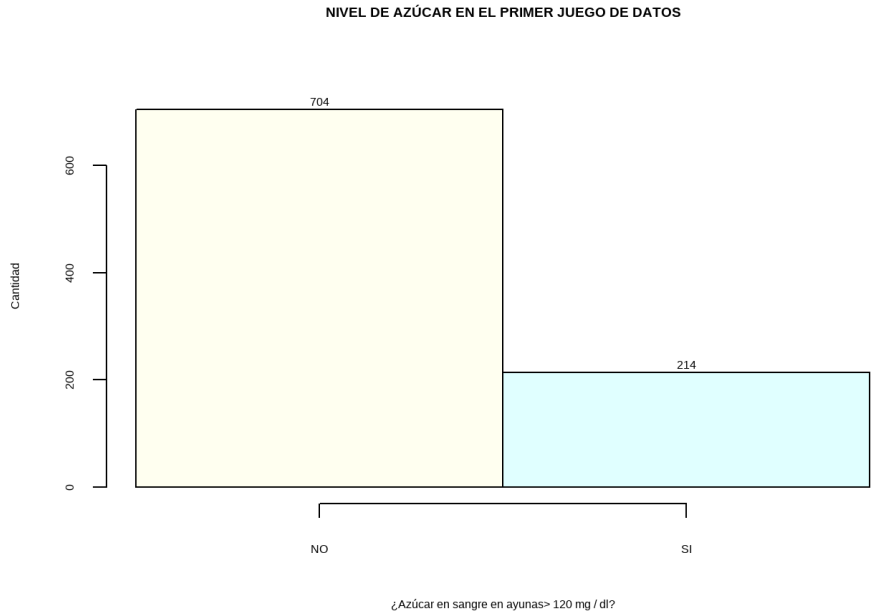


- **NIVEL DE AZÚCAR EN SANGRE EN AYUNAS (FastingBS y fbs)**

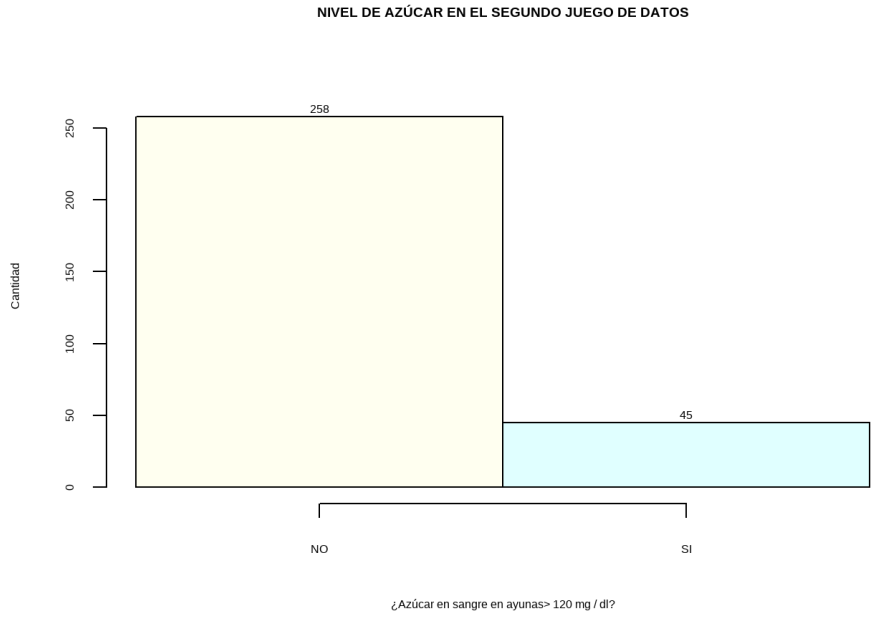
Como se puede comprobar el conjunto de los datos puedes ser 1 o 0, es decir verdadero o falso si se cumple la siguiente condición: si nivel de azúcar en sangre en ayunas> 120 mg / dl.

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

```
#Histograma de la característica Azúcar en sangre en ayunas del primer conjunto de datos
h1 <- hist(datos1$FastingBS, xlab="¿Azúcar en sangre en ayunas> 120 mg / dl?", col=c("ivory", "lightcyan"), ylab="C
antidad", main="NIVEL DE AZÚCAR EN EL PRIMER JUEGO DE DATOS", breaks = 2, ylim = c(0, 750), axes = FALSE)
text(h1$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)
```




```
#Histograma de la característica Azúcar en sangre en ayunas del segundo conjunto de datos
h2 <-hist(datos2$fbs, xlab="¿Azúcar en sangre en ayunas> 120 mg / dl?", col=c("ivory", "lightcyan"), ylab="Cantida
d", main="NIVEL DE AZÚCAR EN EL SEGUNDO JUEGO DE DATOS", breaks = 2, ylim = c(0, 280), axes = FALSE)
text(h2$mids,h2$counts,labels=h2$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75), cex.axis=1, labels = c("NO","SI" ))
axis(2)
```



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

- **ECG EN REPOSO (RestingECG y restecg)**

En el primer conjunto de datos tenemos diferentes parámetros que esta característica puede tomar:

```
+ 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.
```

En el segundo conjunto de datos, los diferentes parámetros que esta característica puede tomas son:

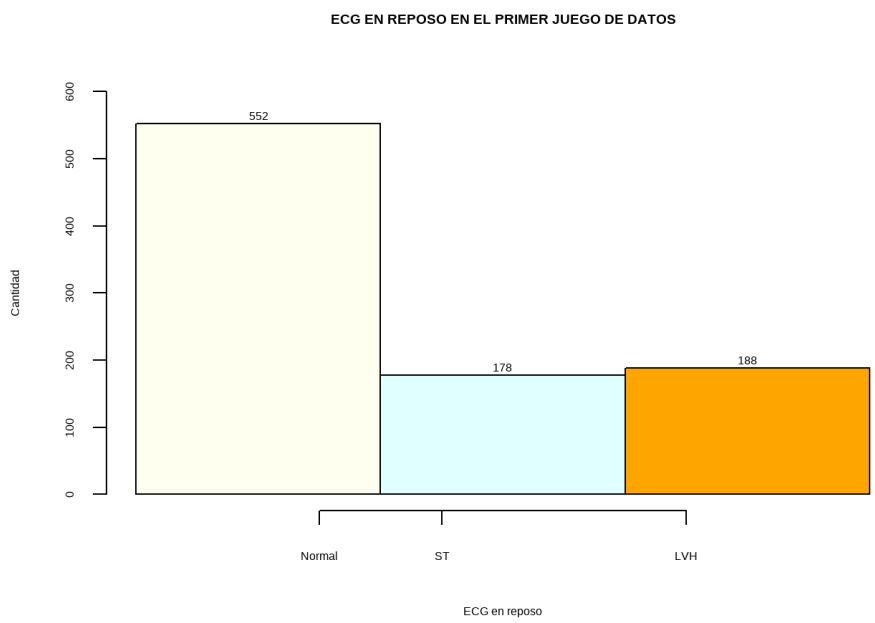
```
+ 0 = normal
+ 1 = con anomalía de la onda ST-T
+ 2 = mostrando hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes.
```

Para normalizar los dos conjuntos de datos, se cambiará los valores del primer conjunto de datos para que sean equivalentes al segundo.

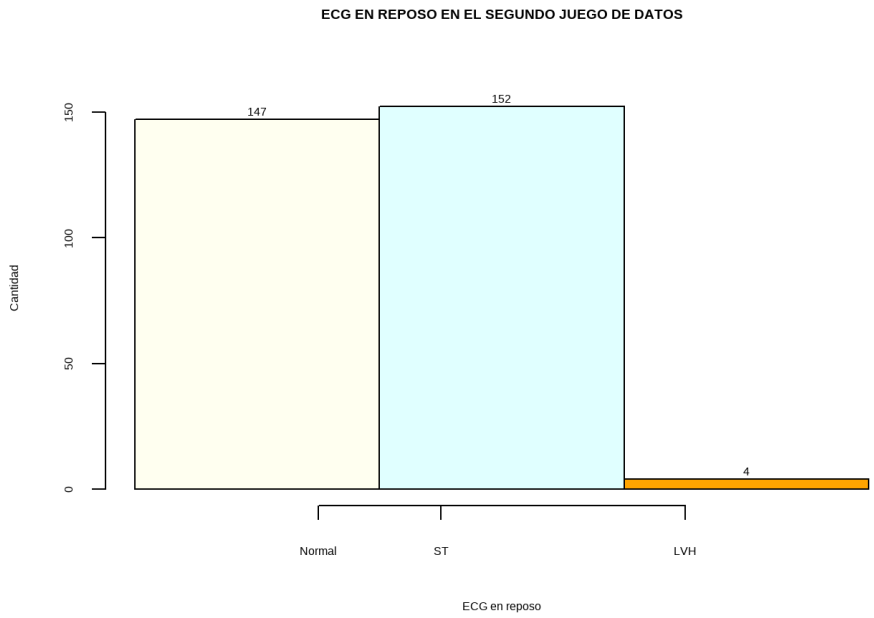
```
#Cambiamos las letras por los números
datos1$RestingECG [datos1$RestingECG == "Normal"]  <- 0
datos1$RestingECG [datos1$RestingECG == "ST"] <- 1
datos1$RestingECG [datos1$RestingECG == "LVH"] <- 2

#Pasamos de carácter a numérico
datos1$RestingECG <- as.numeric(datos1$RestingECG)
```

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

```
#Histograma de la característica ECG en reposo del primer conjunto de datos
h1 <- hist(datos1$RestingECG, xlab="ECG en reposo", col= c("ivory", "lightcyan", "ORANGE"), ylab="Cantidad", main=
"ECG EN REPOSO EN EL PRIMER JUEGO DE DATOS", ylim = c(0, 600), axes = FALSE, breaks=seq(min(datos1$RestingECG)-0.5,
max(datos1$RestingECG)+0.5, by=1) )
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75, 1.75 ), cex.axis=1, labels = c("Normal","ST", "LVH"))
axis(2)
```




```
#Histograma de la característica ECG en reposo del segundo conjunto de datos
h1 <- hist(datos2$restecg, xlab="ECG en reposo", col= c("ivory", "lightcyan", "ORANGE"), ylab="Cantidad", main="ECG
EN REPOSO EN EL SEGUNDO JUEGO DE DATOS", ylim = c(0, 160), axes = FALSE,breaks=seq(min(datos2$restecg)-0.5, max(dat
os2$restecg)+0.5, by=1) )
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75, 1.75 ), cex.axis=1, labels = c("Normal","ST", "LVH"))
axis(2)
```

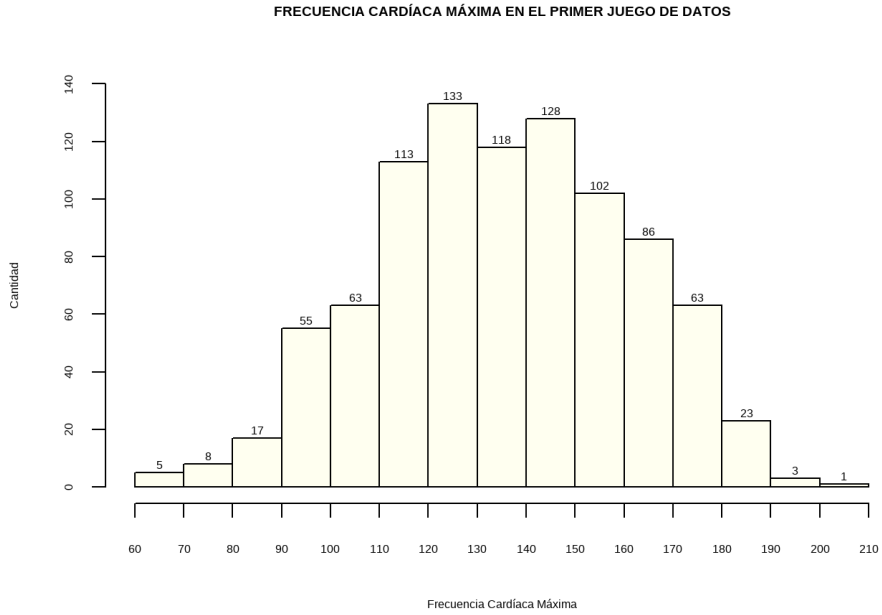


Como se puede contemplar en el primer conjunto de datos los valores HVI es el segundo grupo con más registros, y en el segundo supone un conjunto muy bajo de todas las muestras mientras que las otras dos opciones están muy igualadas.

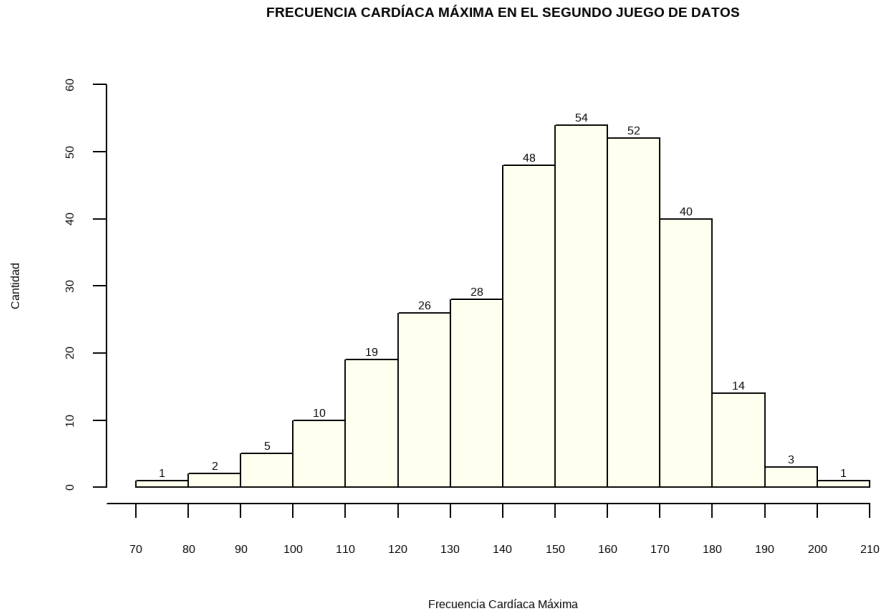
- **FRECUENCIA CARDÍACA MÁXIMA (MaxHR, thalach)**

Dicha característica es de carácter numérica y en el primer conjunto de datos contempla valores desde el 60 al 202 y en el segundo desde el 71 hasta el 202.

```
#Histograma de la característica Frecuencia Cardíaca Máxima del primer conjunto de datos
h1 <- hist(datos1$MaxHR, xlab="Frecuencia Cardíaca Máxima", col="ivory", ylab="Cantidad", main="FRECUENCIA CARDÍACA
MÁXIMA EN EL PRIMER JUEGO DE DATOS", ylim = c(0,140), axes = FALSE)
text(h1$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)
```




```
#Histograma de la característica Frecuencia Cardíaca Máxima del segundo conjunto de datos
h1 <- hist(datos2$thalach, xlab="Frecuencia Cardíaca Máxima", col="ivory", ylab="Cantidad", main="FRECUENCIA CARDÍA
CA MÁXIMA EN EL SEGUNDO JUEGO DE DATOS", ylim = c(0,60), axes = FALSE)
text(h1$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 los dos conjuntos 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, exang)**

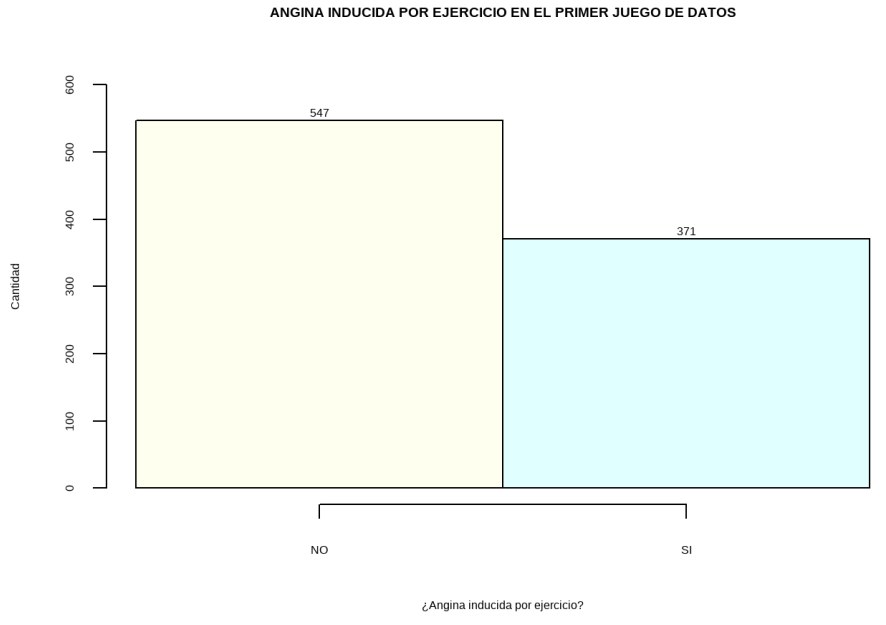
En el primer conjunto de datos tiene los valores Y: Sí, N: No, mientras que en el segundo 1 = sí; 0 = no.

Al igual que se ha hecho con otras características, se normalizará el primer conjunto a favor del segundo conjunto de datos.

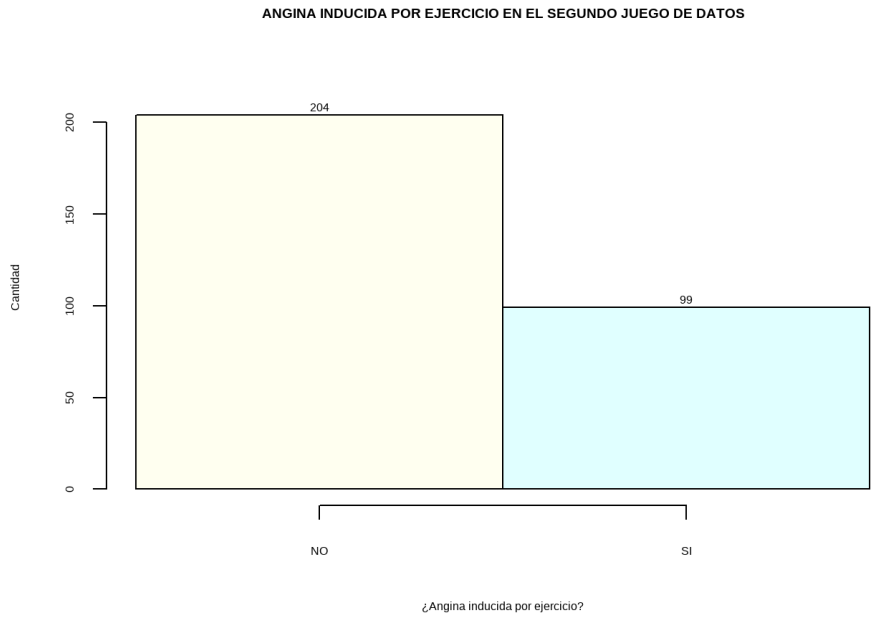
```
#Cambiamos las letras por los números
datos1$ExerciseAngina [datos1$ExerciseAngina == "N"]  <- 0
datos1$ExerciseAngina [datos1$ExerciseAngina == "Y"]  <- 1

#Pasamos de carácter a numérico
datos1$ExerciseAngina <- as.numeric(datos1$ExerciseAngina)
```

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

```
#Histograma de la característica Angina inducida por ejercicio del primer conjunto de datos
h1 <- hist(datos1$ExerciseAngina, xlab="¿Angina inducida por ejercicio?", col=c("ivory", "lightcyan"), ylab="Cantid
ad", main="ANGINA INDUCIDA POR EJERCICIO EN EL PRIMER JUEGO DE DATOS", breaks = 2, ylim = c(0, 600), axes = FALSE)
text(h1$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)
```



```
#Histograma de la característica Angina inducida por ejercicio del segundo conjunto de datos
h2 <-hist(datos2$exang, xlab="¿Angina inducida por ejercicio?", col=c("ivory", "lightcyan"), ylab="Cantidad", main=
"ANGINA INDUCIDA POR EJERCICIO EN EL SEGUNDO JUEGO DE DATOS", breaks = 2, ylim = c(0, 220), axes = FALSE)
text(h2$mids,h2$counts,labels=h2$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75), cex.axis=1, labels = c("NO","SI" ))
axis(2)
```

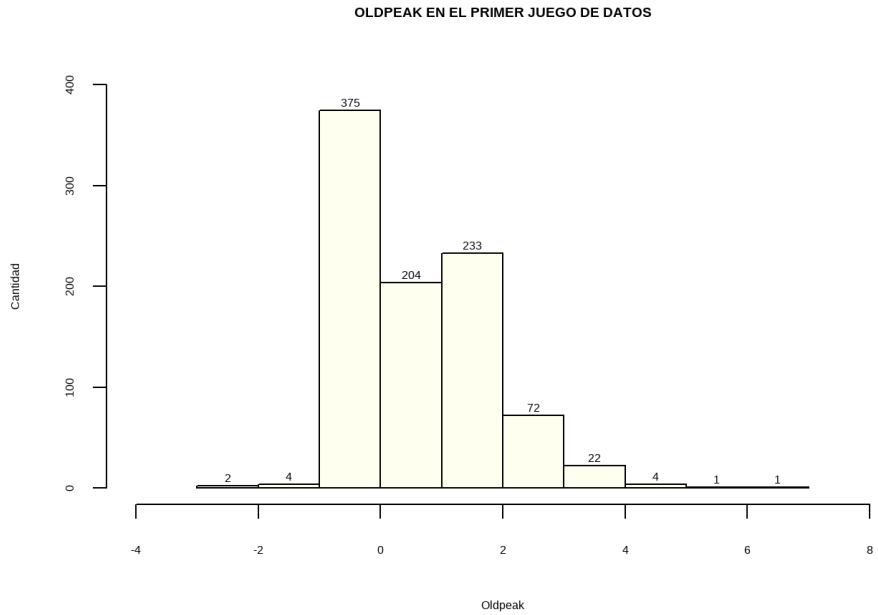


Como se puede apreciar, hay mas casos en que NO se ha producido una angina inducida por el ejercicio de que Si se haya producido en los dos conjuntos de datos.

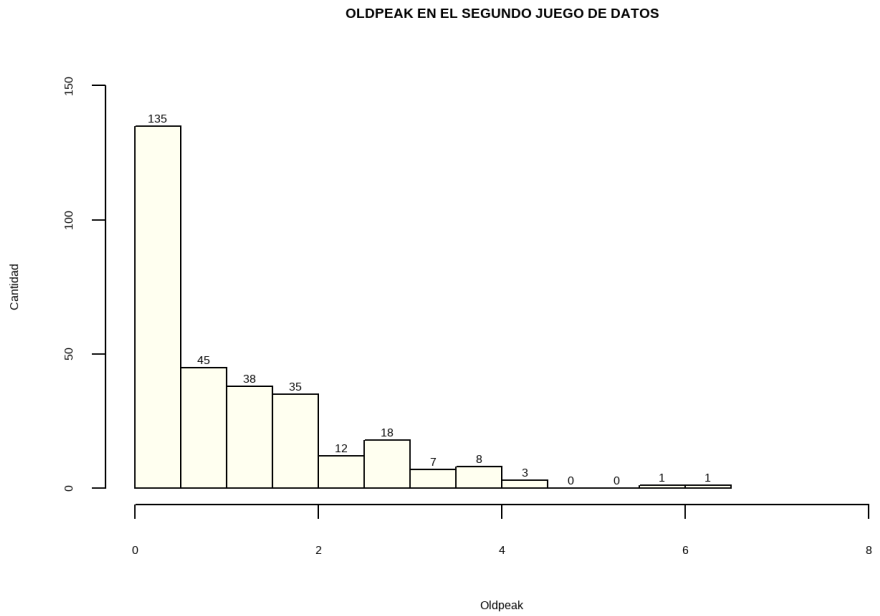
- **OLDPEAK**

Esta característica de tipo numérica puede abarcar valores negativos hasta (en el caso del primer conjunto) hasta un máximo de un valor igual a 6,2 (en ambos conjuntos de datos)

```
#Histograma de la característica Oldpeak del primer conjunto de datos
h1 <- hist(datos1$Oldpeak, xlab="Oldpeak", col="ivory", ylab="Cantidad", main="OLDPEAK EN EL PRIMER JUEGO DE DATOS"
, ylim = c(0,400), xlim = c(-4, 8))
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
```



```
#Histograma de la característica Oldpeak del segundo conjunto de datos
h1 <- hist(datos2$oldpeak, xlab="Oldpeak", col="ivory", ylab="Cantidad", main="OLDPEAK EN EL SEGUNDO JUEGO DE DATO
S", ylim = c(0,150), xlim = c(0, 8))
text(h1$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 en el primer caso mientras que en el segundo juego de datos los valores iniciales tienen mas muestras. Observar que en el segundo conjunto tiene rango de valores positivos, mientras que en el primer conjunto de datos abarca un rango mas amplio.

- **PENDIENTE DEL SEGMENTO ST (ST_Slope, pendiente)**

Como ocurría en otras características anteriores cada conjunto de datos los mide de una manera distinta, siendo en el primer conjunto:

```
+ Up: uploping
+ Flat: flat
+ Down: downsloping
```

Y en el segundo conjunto de datos:

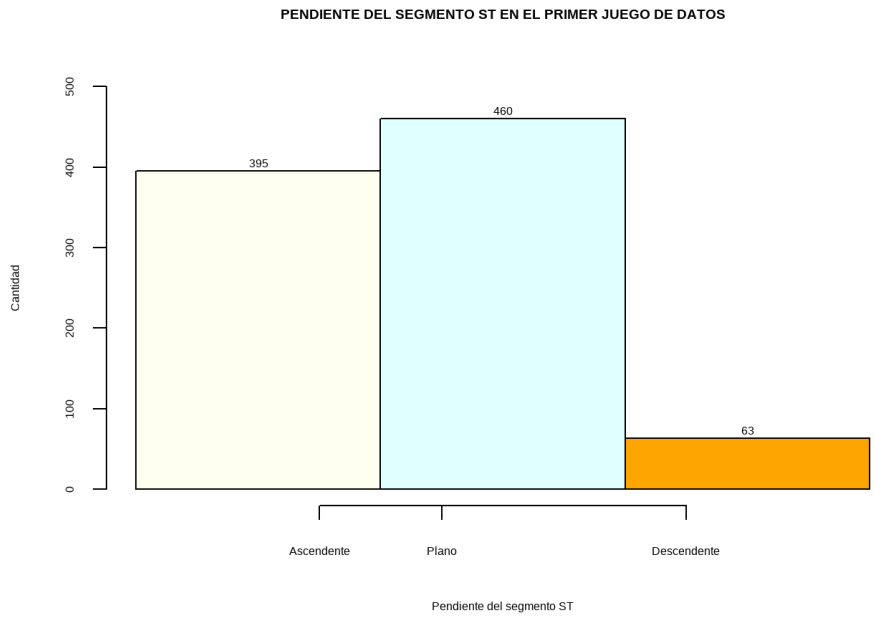
```
+ Valor 0: pendiente ascendente
+ Valor 1: plano
+ Valor 2: pendiente descendente
```

Y como se ha realizado antes, se normalizará el primer conjunto a favor del segundo.

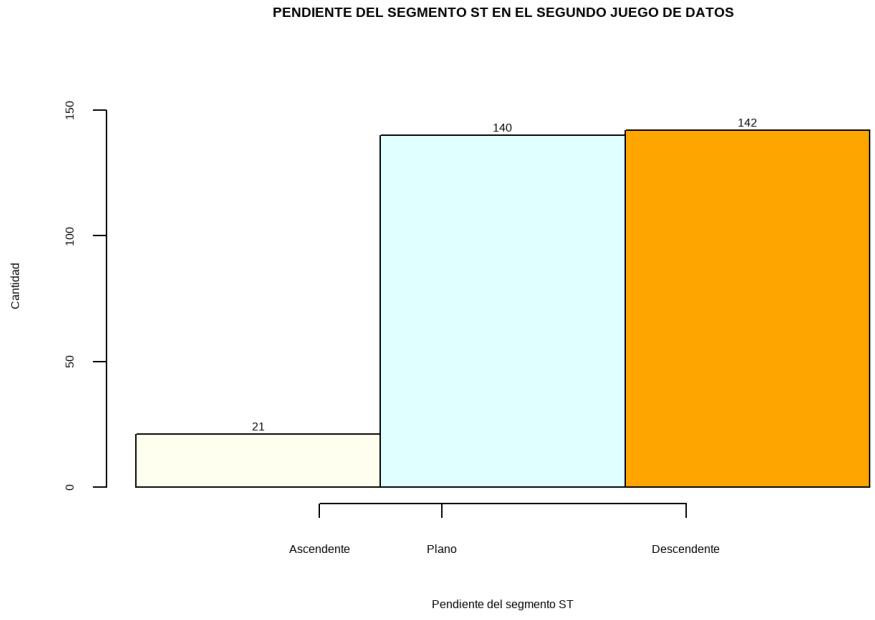
```
#Cambiamos las letras por los números
datos1$ST_Slope [datos1$ST_Slope == "Up"]   <- 0
datos1$ST_Slope [datos1$ST_Slope == "Flat"] <- 1
datos1$ST_Slope [datos1$ST_Slope == "Down"] <- 2

#Pasamos de carácter a numérico
datos1$ST_Slope <- as.numeric(datos1$ST_Slope)
```

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

```
#Histograma de la característica Pendiente del segmento ST del primer conjunto de datos
h1 <- hist(datos1$ST_Slope, xlab="Pendiente del segmento ST", col= c("ivory", "lightcyan", "ORANGE"), ylab="Cantida
d", main="PENDIENTE DEL SEGMENTO ST EN EL PRIMER JUEGO DE DATOS", ylim = c(0, 500), axes = FALSE,breaks=seq(min(dat
os1$ST_Slope)-0.5, max(datos1$ST_Slope)+0.5, by=1) )
text(h1$mids,h1$counts,labels=h1$counts, adj=c(0.5, -0.5))
axis(1, at =c(0.25, 0.75,1.75), cex.axis=1, labels = c("Ascendente","Plano", "Descendente"))
axis(2)
```




```
#Histograma de la característica Pendiente del segmento ST del segundo conjunto de datos
h1 <- hist(datos2$slope, xlab="Pendiente del segmento ST", col= c("ivory", "lightcyan", "ORANGE"), ylab="Cantidad",
main="PENDIENTE DEL SEGMENTO ST EN EL SEGUNDO JUEGO DE DATOS", ylim = c(0, 160), axes = FALSE,breaks=seq(min(datos2
$slope)-0.5, max(datos2$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, 0.75,1.75), cex.axis=1, labels = c("Ascendente","Plano", "Descendente"))
axis(2)
```

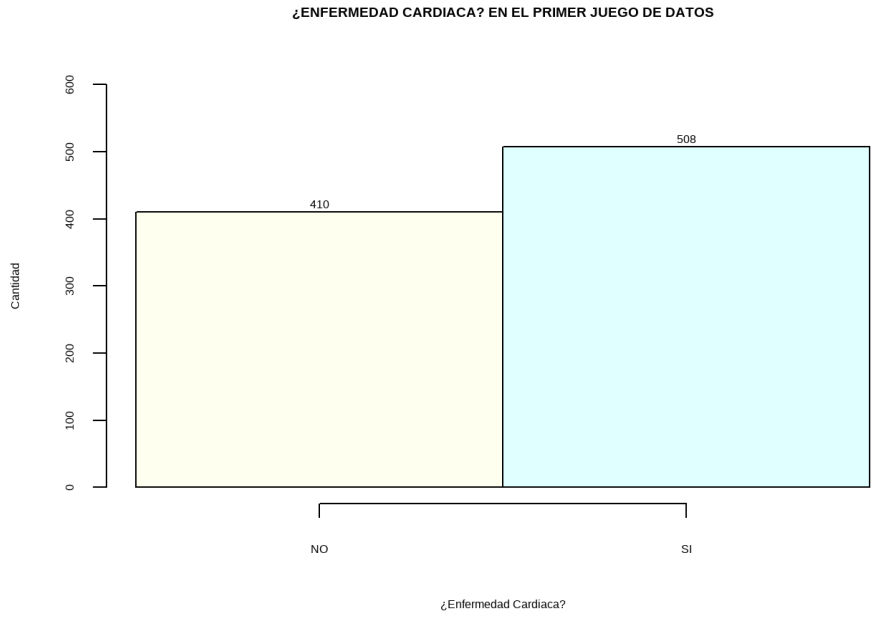


El caso más común de ambos conjuntos es que la pendiente sea plana, sin embargo en el primer conjunto la tendencia del segundo caso más común es ascendente y en el segundo conjunto descendente. Esto es bastante bueno ya que nos permite tener una visión mas amplia de todos los tipos de pendientes.

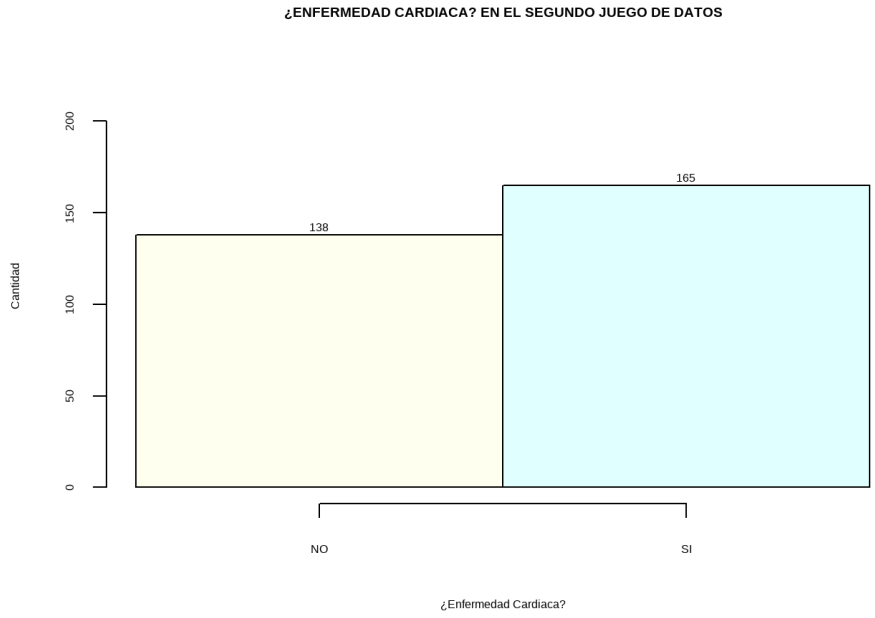
- **¿ENFERMEDAD CARDIACA? (HeartDisease, target)**

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

```
#Histograma de la característica¿Enfermedad Cardiaca? del primer conjunto de datos
h1 <- hist(datos1$HeartDisease, xlab="¿Enfermedad Cardiaca?", col=c("ivory", "lightcyan"), ylab="Cantidad", main=
"¿ENFERMEDAD CARDIACA? EN EL PRIMER JUEGO DE DATOS", breaks = 2, ylim = c(0, 600), axes = FALSE)
text(h1$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)
```




```
#Histograma de la característica ¿Enfermedad Cardiaca? del segundo conjunto de datos
h2 <-hist(datos2$target, xlab="¿Enfermedad Cardiaca?", col=c("ivory", "lightcyan"), ylab="Cantidad", main="¿ENFERME
DAD CARDIACA? EN EL SEGUNDO JUEGO DE DATOS", breaks = 2, ylim = c(0, 220), axes = FALSE)
text(h2$mids,h2$counts,labels=h2$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.

2.3.3 Tratamiento características distintas

Antes se ha mencionado que el segundo conjunto tiene dos características presentes que el primer conjunto no tiene. Haciendo un análisis de las estadísticas y la definición de dada al principio de cada una de las dos características, se ha decidió descartarlas por las siguientes razones:

```
 + Si se quieren tener en cuenta tendremos casi el 75 % de valores nulos.
 + En el caso del numero de vasos afectado (CA) es algo especial para cada caso y no se puede obtener similitudes c
on otros casos.
 + La Talasemia tampoco se puede calcular la cantidad que tiene a través de otros campos.
```

Una vez que tenemos todas las características de los dos conjuntos con la misma normalización, se va a juntar los dos conjuntos de datos en uno solo.

2.4 Construcción de conjunto de datos final

2.4.1 Normalización de nombres de columnas

Lo primero es la normalización de los nombre de las columnas de los dos conjunto de datos

```
#Obtenemos el nombre de las columnas del primer conjunto de datos
colnames(datos1)
```

```
##  [1] "Age"            "Sex"            "ChestPainType"  "RestingBP"      "Cholesterol"    "FastingBS"      "Rest
ingECG"     "MaxHR"          "ExerciseAngina"
## [10] "Oldpeak"        "ST_Slope"       "HeartDisease"
```

```
#Renombramos las columnas del primer conjunto de datos

colnames(datos1)[1]<-  "EDAD"
colnames(datos1)[2]<-  "SEXO"
colnames(datos1)[3]<-  "TIPO DOLOR TORAX"
colnames(datos1)[4]<-  "PRESIÓN ARTERIAL"
colnames(datos1)[5]<-  "CORESTEROL"
colnames(datos1)[6]<-  "NIVEL DE AZÚCAR"
colnames(datos1)[7]<-  "ECG EN REPOSO"
colnames(datos1)[8]<-  "FREC CARDÍACA MÁX"
colnames(datos1)[9]<-  "ANGINA x EJERCICIO"
colnames(datos1)[10]<- "OLDPEAK"
colnames(datos1)[11]<- "PENDIENTE ST"
colnames(datos1)[12]<- "E. CARDIACA"


#Vemos el nombre de las columnas del primer conjunto de datos
colnames(datos1)
```

```
##  [1] "EDAD"               "SEXO"               "TIPO DOLOR TORAX"   "PRESIÓN ARTERIAL"   "CORESTEROL"         "N
IVEL DE AZÚCAR"    "ECG EN REPOSO"
##  [8] "FREC CARDÍACA MÁX"  "ANGINA x EJERCICIO" "OLDPEAK"            "PENDIENTE ST"       "E. CARDIACA"
```

```
#Obtenemos el nombre de las columnas del segundo conjunto de datos
colnames(datos2)
```

```
##  [1] "ï..age"   "sex"      "cp"       "trestbps" "chol"     "fbs"      "restecg"  "thalach"  "exang"    "oldpea
k"  "slope"    "ca"       "thal"     "target"
```

```
#Renombramos las columnas del primer segundo de datos

colnames(datos2)[1]<-  "EDAD"
colnames(datos2)[2]<-  "SEXO"
colnames(datos2)[3]<-  "TIPO DOLOR TORAX"
colnames(datos2)[4]<-  "PRESIÓN ARTERIAL"
colnames(datos2)[5]<-  "CORESTEROL"
colnames(datos2)[6]<-  "NIVEL DE AZÚCAR"
colnames(datos2)[7]<-  "ECG EN REPOSO"
colnames(datos2)[8]<-  "FREC CARDÍACA MÁX"
colnames(datos2)[9]<-  "ANGINA x EJERCICIO"
colnames(datos2)[10]<- "OLDPEAK"
colnames(datos2)[11]<- "PENDIENTE ST"
colnames(datos2)[14]<- "E. CARDIACA"

#Eliminamos las colunmas que no vamos a usar
datos2$ca <- NULL
datos2$thal <- NULL

#Vemos el nombre de las columnas del primer conjunto de datos
colnames(datos2)
```

```
##  [1] "EDAD"               "SEXO"               "TIPO DOLOR TORAX"   "PRESIÓN ARTERIAL"   "CORESTEROL"         "N
IVEL DE AZÚCAR"    "ECG EN REPOSO"
##  [8] "FREC CARDÍACA MÁX"  "ANGINA x EJERCICIO" "OLDPEAK"            "PENDIENTE ST"       "E. CARDIACA"
```

2.4.2 Fusión de los conjuntos de datos

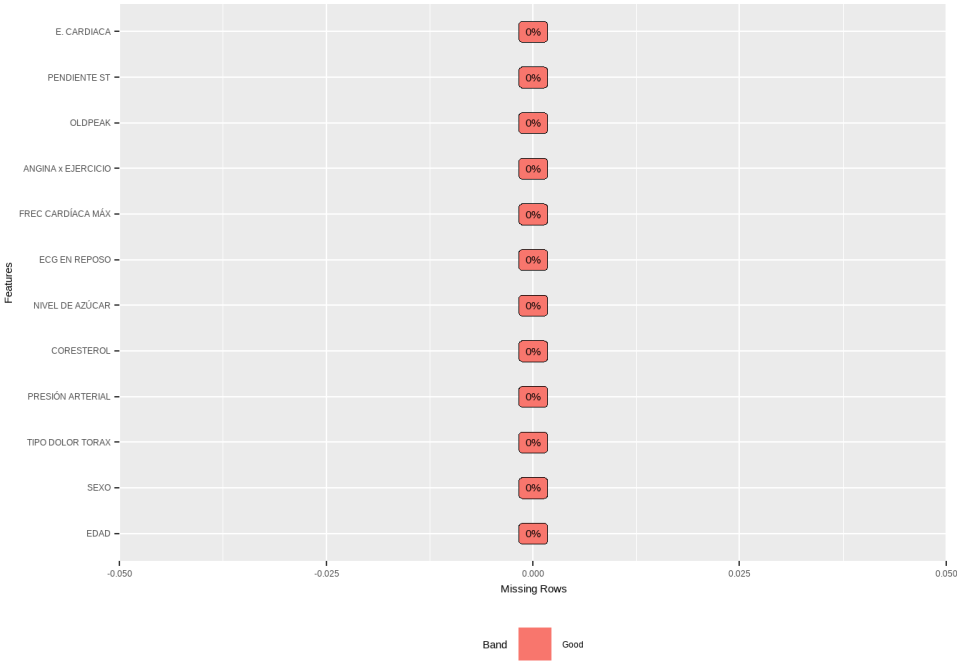
```
#Fusionamos los dos conjuntos de datos
datos_final <- merge(x=datos1, y=datos2, all = TRUE)

#Verificamos la estructura del segundo juego
str(datos_final)
```

```
## 'data.frame':    1221 obs. of  12 variables:
##  $ EDAD              : int  28 29 29 29 29 30 31 31 32 32 ...
##  $ SEXO              : num  1 1 1 1 1 0 0 1 0 1 ...
##  $ TIPO DOLOR TORAX  : num  1 1 1 1 1 0 1 3 1 0 ...
##  $ PRESIÓN ARTERIAL  : int  130 120 130 130 140 170 100 120 105 95 ...
##  $ CORESTEROL        : num  132 243 204 204 263 237 219 270 198 198 ...
##  $ NIVEL DE AZÚCAR   : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ ECG EN REPOSO     : num  2 0 0 2 0 1 1 0 0 0 ...
##  $ FREC CARDÍACA MÁX : int  185 160 202 202 170 170 150 153 165 127 ...
##  $ ANGINA x EJERCICIO: num  0 0 0 0 0 0 0 1 0 0 ...
##  $ OLDPEAK           : num  0 0 0 0 0 0 0 1.5 0 0.7 ...
##  $ PENDIENTE ST      : num  0 0 2 0 0 0 0 1 0 0 ...
##  $ E. CARDIACA       : int  0 0 1 0 0 0 0 1 0 1 ...
```

```
#Estadísticas básicas
summary(datos_final)
```

```
##       EDAD            SEXO        TIPO DOLOR TORAX PRESIÓN ARTERIAL   CORESTEROL    NIVEL DE AZÚCAR  ECG EN REPO
SO    FREC CARDÍACA MÁX ANGINA x EJERCICIO
##  Min.   :28.00   Min.   :0.0000   Min.   :0.000    Min.   : 80.0    Min.   : 85.0   Min.   :0.0000   Min.   :0.0
000   Min.   : 60       Min.   :0.0000
##  1st Qu.:47.00   1st Qu.:1.0000   1st Qu.:1.000    1st Qu.:120.0    1st Qu.:198.0   1st Qu.:0.0000   1st Qu.:0.0
000   1st Qu.:122       1st Qu.:0.0000
##  Median :54.00   Median :1.0000   Median :2.000    Median :130.0    Median :228.0   Median :0.0000   Median :0.0
000   Median :141       Median :0.0000
##  Mean   :53.72   Mean   :0.7633   Mean   :1.933    Mean   :132.3    Mean   :238.5   Mean   :0.2121   Mean   :0.5
848   Mean   :140       Mean   :0.3849
##  3rd Qu.:60.00   3rd Qu.:1.0000   3rd Qu.:3.000    3rd Qu.:140.0    3rd Qu.:269.0   3rd Qu.:0.0000   3rd Qu.:1.0
000   3rd Qu.:160       3rd Qu.:1.0000
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000    Max.   :200.0    Max.   :603.0   Max.   :1.0000   Max.   :2.0
000   Max.   :202       Max.   :1.0000
##     OLDPEAK         PENDIENTE ST     E. CARDIACA
##  Min.   :-2.6000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 0.6000   Median :1.0000   Median :1.0000
##  Mean   : 0.9251   Mean   :0.8272   Mean   :0.5512
##  3rd Qu.: 1.6000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   : 6.2000   Max.   :2.0000   Max.   :1.0000
```

```
#Comprobar valores nulos
plot_missing(datos_final)
```



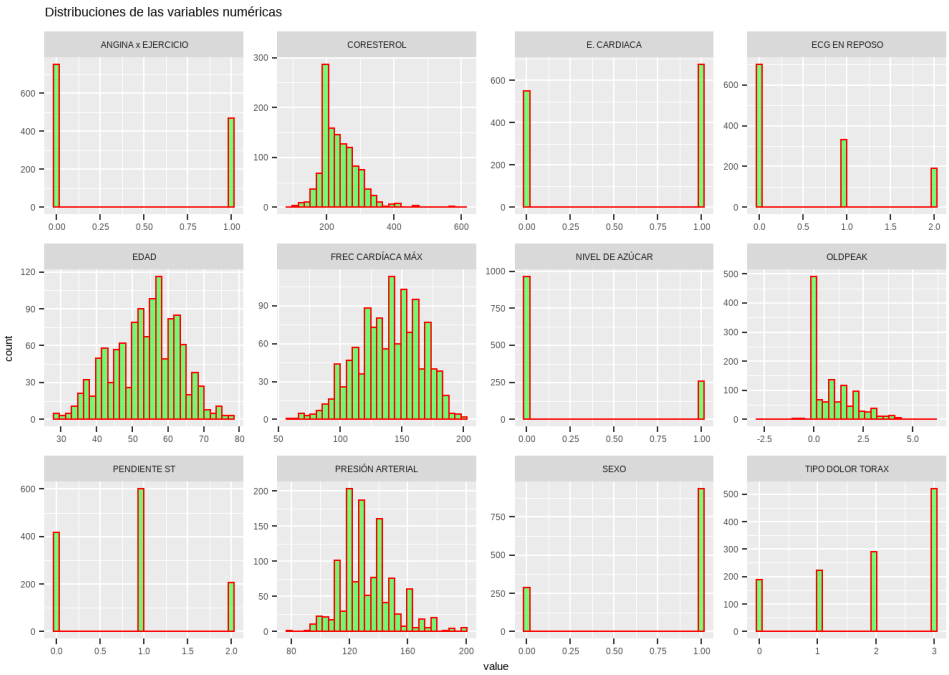
Podemos concluir que el nuevo juego de datos tiene las siguientes características:

- **Edad:** la edad de la persona en años.
- **Sexo:** el sexo de la persona. Los valores que puede tomar son:
    - 1 = hombre.
    - 0 = mujer.
- **Tipo de dolor Torax:** tipo de dolor torácico experimentado. Los valores que puede tomar son:
    - 0 = Angina típica.
    - 1 = Angina atípica.
    - 2 = Dolor no anginoso.
    - 3 = Asintomático.
- **Presión arterial:** la presión arterial en reposo de la persona (medido en mm/Hg) al ingreso en el hospital.
- **Colesterol:** colesterol sérico de la persona [medido eb mm/dl]
- **Nivel de azúcar en sangre:** estando el paciente en ayunas. Los valores que puede tomar son dada la siguiente condición < 120 mg/dl>> son:
    - 1 = Verdadero.
    - 0 = Falso.
- **ECG en reposo:** resultados del electrocardiograma en reposo. Los valores que puede tomar son:
    - 0 = Normal.
    - 1 = Con anomalía de la onda ST-T.
    - 2 = Mostrando hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes.
- **Frec cardíaca max:** frecuencia cardíaca máxima alcanzada por la persona.
- **Angina x ejercicio:** si se ha producido una angina al realizar ejercicio. Los valores que puede tomar son:
    - 1 = Sí.
    - 0 = No.
- **Oldpeak:** depresión del ST inducida por el ejercicio en relación con el reposo.
- **Pendiente ST:** la pendiente del segmento ST de ejercicio pico. Los valores que puede tomar son:
    - 0 = Pendiente Ascendente.
    - 1 = Plano.
    - 2 = Pendiente Descendente.
- **¿E. Cardíaca?:** si la persona tiene alguna enfermedad cardíaca. Los valores que puede tomar son:
    - 1 = Sí.
    - 0 = No.

2.4.3 Análisis exploratorio del nuevo conjunto de datos

Una vez descrito el nuevo juego de datos, se va a generar histogramas para verificar la distribución de las variables.

```
library(purrr)
library(tidyr)
library(ggplot2)

datos_final %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram(col="red",
                   fill="green",
                   alpha = 0.5,) +
    ggtitle("Distribuciones de las variables numéricas")
```

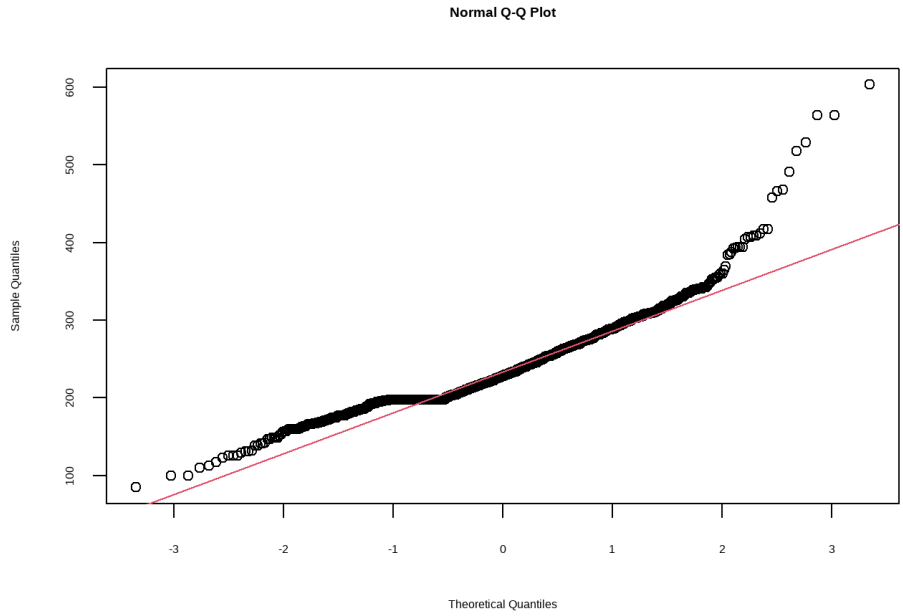


NOMBRE VARIABLE	DISTRIBUCIÓN	EXPLICACIÓN
ANGINA x EJERCICIO	Normal	Hay mas casos en que la angina no ha sido inducida por ejercicio
CORESTEROL	Sesgado a la derecha	Cifras más “bajas” tienen más registros que cifras más altas
E. CARDIACA	Normal	Hay más casos en que se tiene una enfermedad cardiaca
ECG EN REPOSO	Sesgado a la derecha	Casos normales tienen mayor peso que con alguna patología
EDAD	Normal	Hay más casos en personas con mediana edad que en los extremos
FREC CARDÍACA MÁX	Normal	Frecuencia de los datos entre rangos intermedios
NIVEL DE AZÚCAR	Sesgado a la derecha	Hay más casos en que se tiene el nivel bien
OLDPEAK	Sesgado a la derecha	Existe un grupo con una diferencia bastante grande que con el resto de los datos
PENDIENTE ST	Normal	Más casos con pendiente normal que alterada
PRESIÓN ARTERIAL	Normal	Frecuencia de los datos entre rangos intermedios
SEXO	Sesgado a la izquierda	Hay más pacientes hombres que mujeres
TIPO DOLOR TORAX	Sesgado a la izquierda	Hay más casos asintomáticos

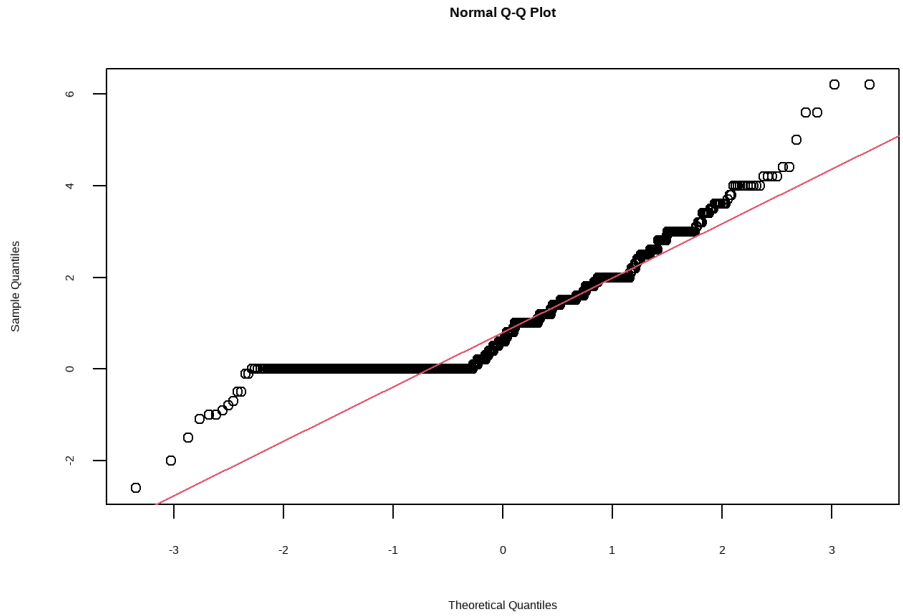
Solo 5 de las doce características tienen una distribución normal, aunque debemos contemplar que alguna de las características está normalizada para tratar dos o tres valores que esas características pueden tener, teniendo en cuenta esto ultimo podemos decir que las únicas dos características que tienen distribuciones distintas son: **Colesterol** y **Oldpeak**.

Para comprobar que estamos en los cierto en las dos características con distribución sesgada (Colesterol y Oldpeak) se va a realizar test de normalidad para verificar y con la “función qqnorm” podríamos hacer un Q-Q plot para ver si una variable determinada tiene una distribución normal.

```
#Colesterol
qqnorm(datos_final$CORESTEROL);qqline(datos_final$CORESTEROL, col = 2)
```

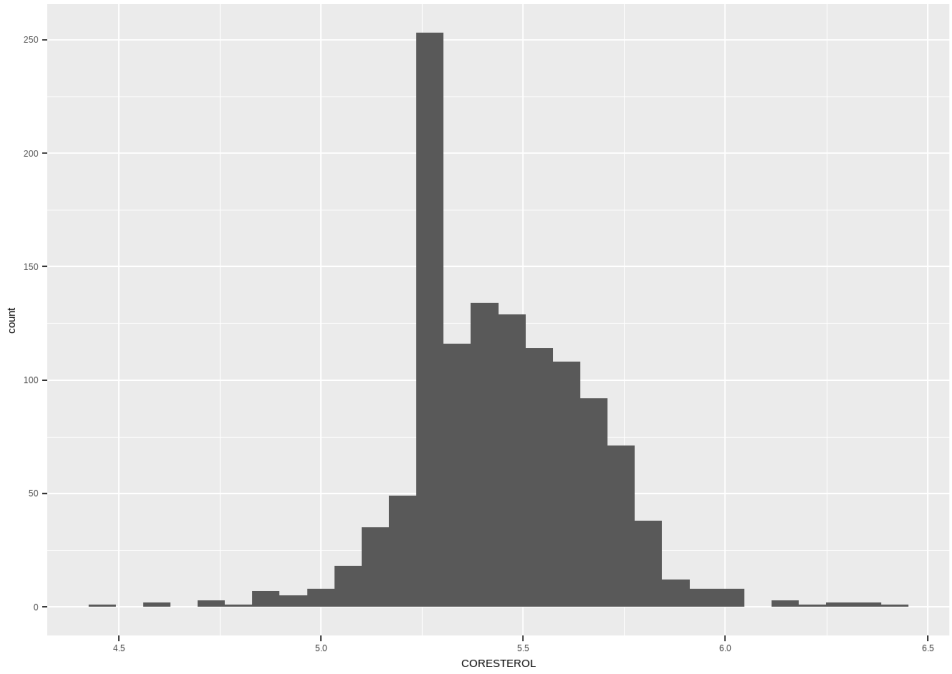



```
#Oldpeak
qqnorm(datos_final$OLDPEAK);qqline(datos_final$OLDPEAK, col = 2)
```

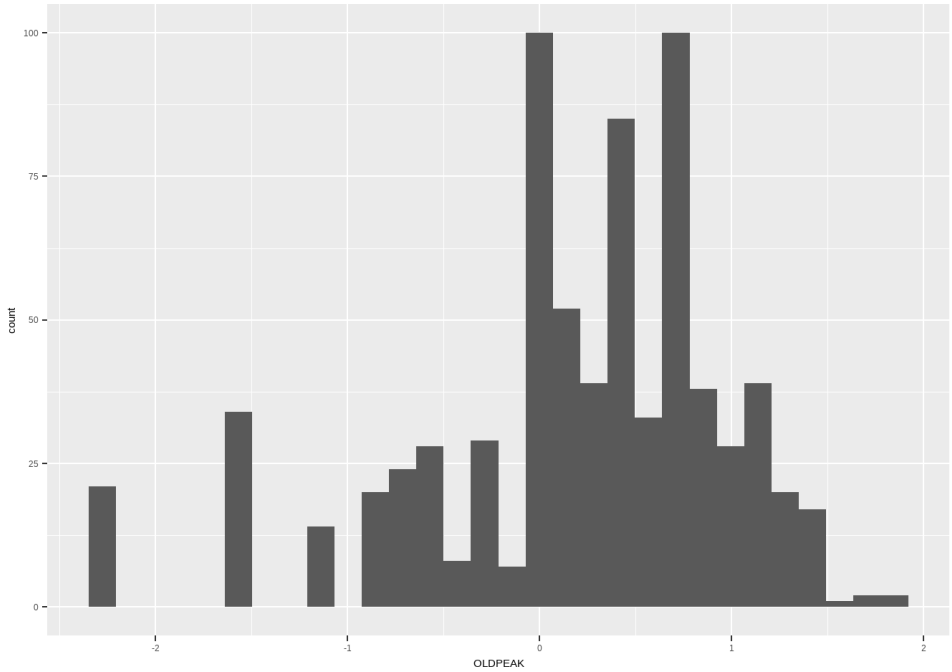


El procedimiento que se puede seguir cuando tenemos una variable que no sigue una distribución normal es la de aplicar el logaritmo a la variable. Lo verificamos de la siguiente manera para las dos características: Colesterol y Oldpeak.

```
#Coresterol
Coresterol_log<- log(datos_final$CORESTEROL)
ggplot(datos_final, aes(x = Coresterol_log)) + geom_histogram() + xlab("CORESTEROL")
```

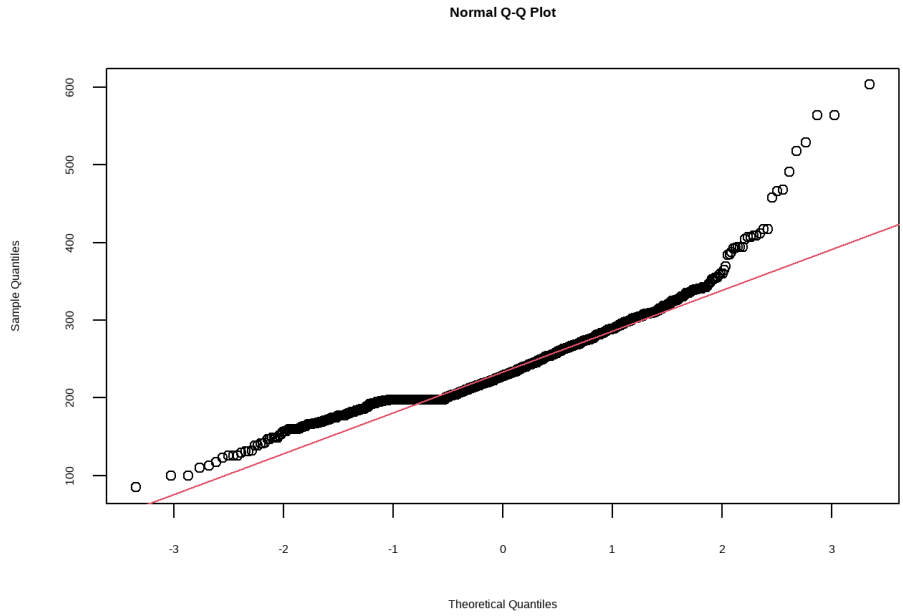



```
#Oldpeak.
Oldpeak_log<- log(datos_final$OLDPEAK)
ggplot(datos_final, aes(x = Oldpeak_log)) + geom_histogram() + xlab("OLDPEAK")
```

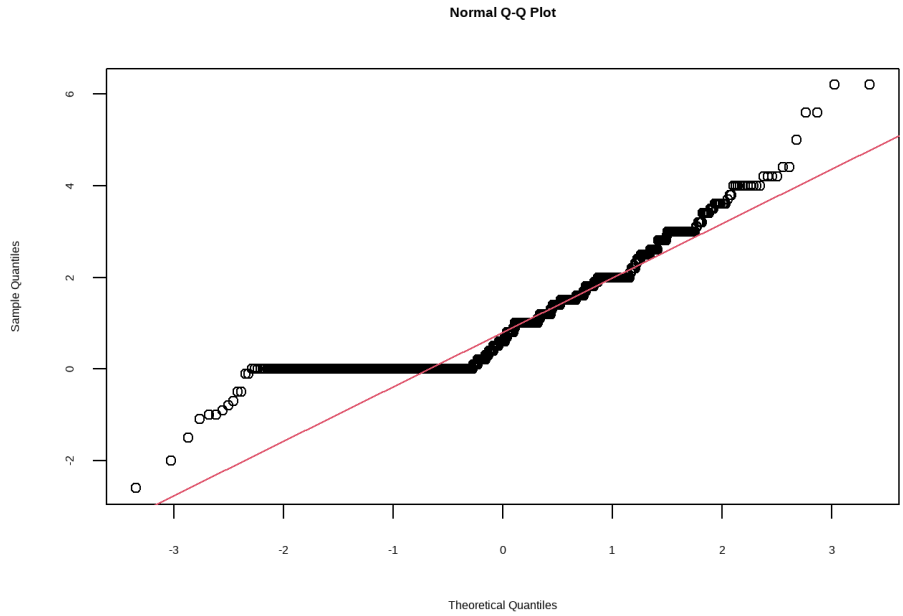


Observamos como ahora cambia las distribuciones. Lo comprobamos con el Q-Q plot para confirmarlo.

```
#Colesterol
qqnorm(datos_final$CORESTEROL);qqline(datos_final$CORESTEROL, col = 2)
```



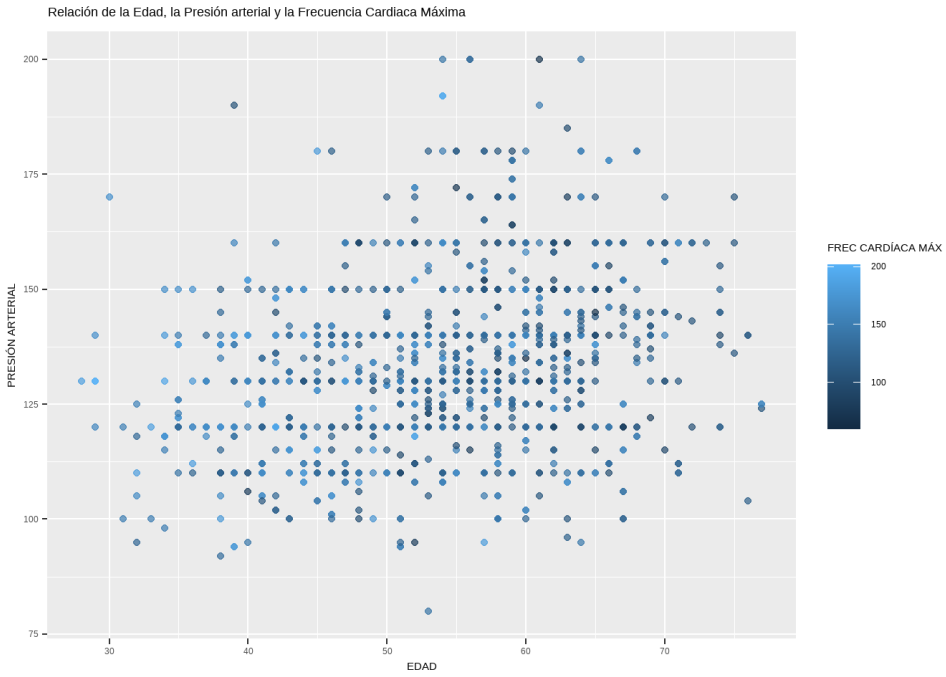

```
#Oldpeak
qqnorm(datos_final$OLDPEAK);qqline(datos_final$OLDPEAK, col = 2)
```



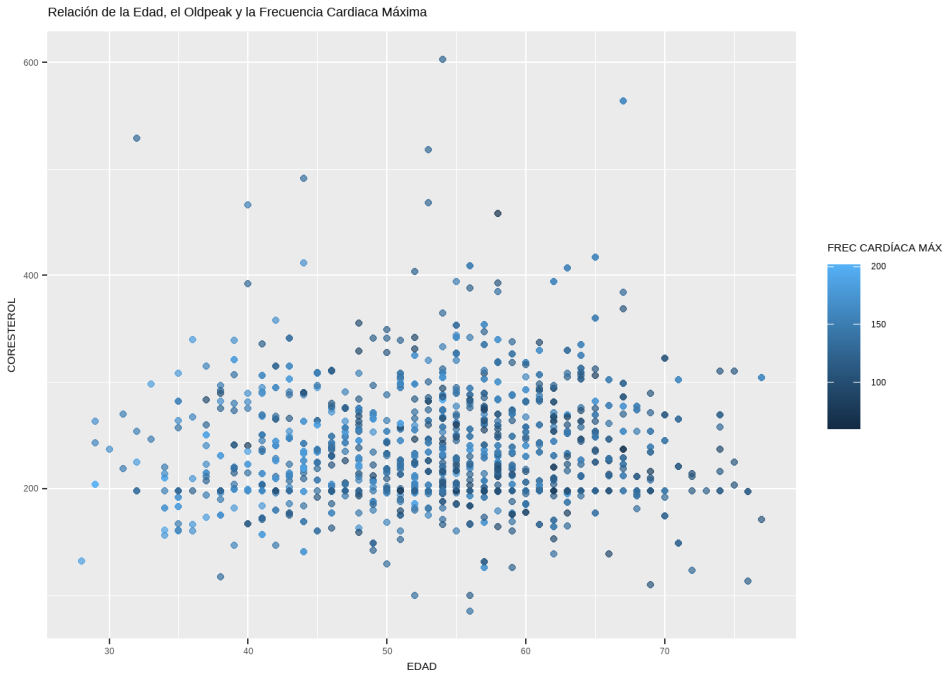
Los test de normalidad no son los esperados, así que se mantendrán estos valores tal y como están. Y continuamos con el análisis exploratorio del nuevo conjunto de datos.

A continuación, se va a representar los niveles de ciertas características en relación con otras

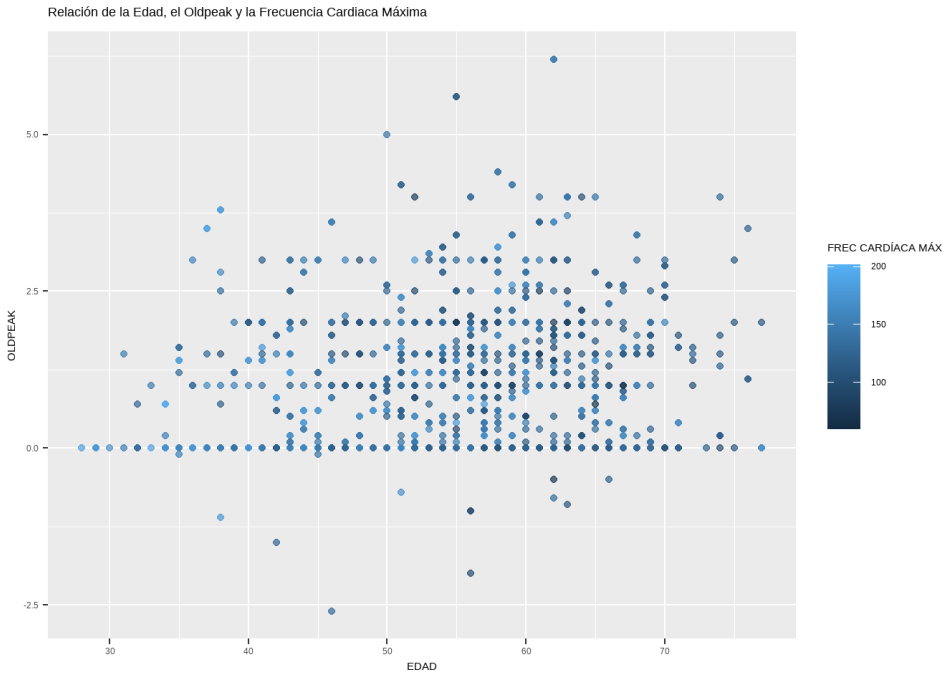
```
#Relación de la Edad, la Presión arterial y la Frecuencia Cardiaca Máxima.
datos_final %>%
  ggplot(aes(x=EDAD,y=`PRESIÓN ARTERIAL`,color=`FREC CARDÍACA MÁX`))+
  geom_point(alpha=0.7)+xlab("EDAD") +
  ylab("PRESIÓN ARTERIAL")+
  ggtitle("Relación de la Edad, la Presión arterial y la Frecuencia Cardiaca Máxima")
```




```
#Relación de la Edad, el Coresterol y la Frecuencia Cardiaca máxima.
datos_final %>%
  ggplot(aes(x=EDAD,y=CORESTEROL,color=`FREC CARDÍACA MÁX`))+
  geom_point(alpha=0.7)+xlab("EDAD") +
  ylab("CORESTEROL")+
  ggtitle("Relación de la Edad, el Oldpeak y la Frecuencia Cardiaca Máxima")
```



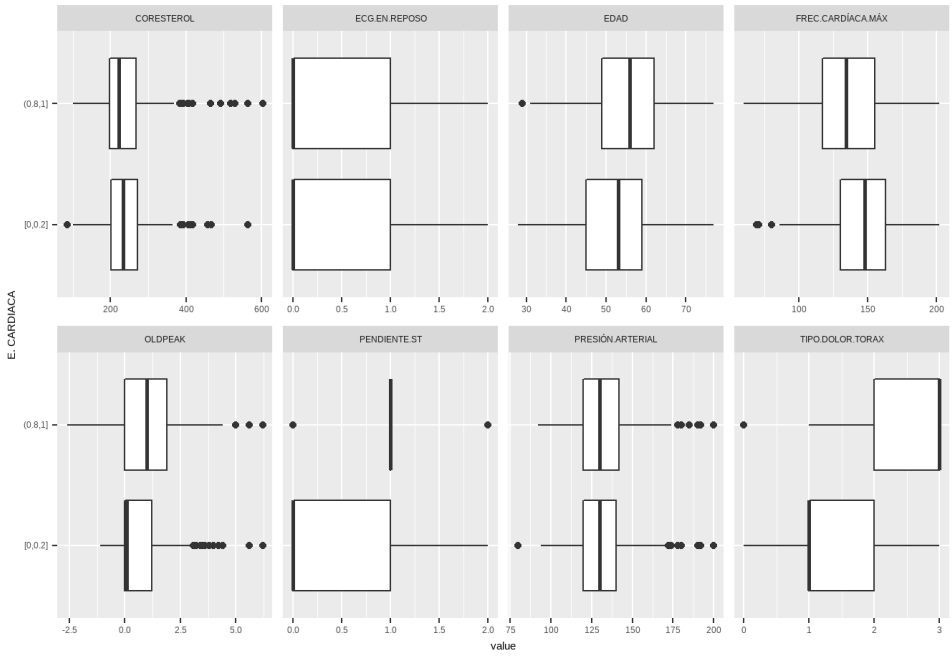

```
#Relación de la Edad, el Coresterol y la Frecuencia Cardiaca máxima.
datos_final %>%
  ggplot(aes(x=EDAD,y=OLDPEAK,color=`FREC CARDÍACA MÁX`))+
  geom_point(alpha=0.7)+xlab("EDAD") +
  ylab("OLDPEAK")+
  ggtitle("Relación de la Edad, el Oldpeak y la Frecuencia Cardiaca Máxima")
```



Gracias a estas representaciones se pueden ver las relaciones entre unas características y otras.

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.

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



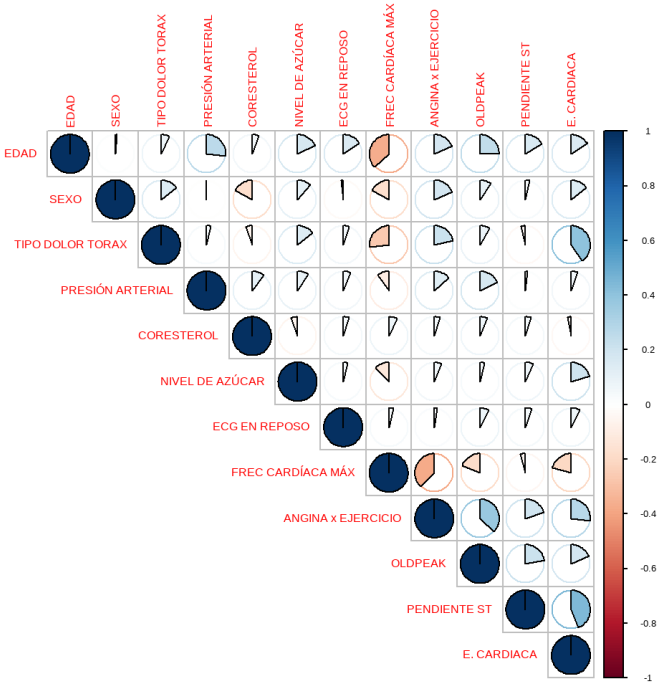
2.4.4 Correlaciones

Una vez realizado el análisis exploratorio, se va a realizar las correlaciones de las características

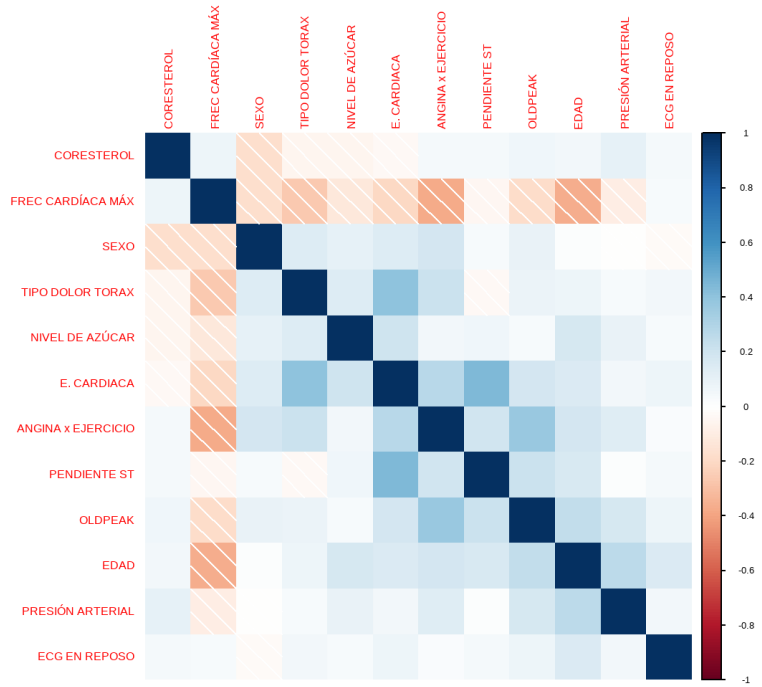
```
#Calculamos las correlaciones
cor_datos <- cor(datos_final)
cor_datos
```

```
##                              EDAD         SEXO TIPO DOLOR TORAX PRESIÓN ARTERIAL  CORESTEROL NIVEL DE AZÚCAR ECG EN
REPOSO FREC CARDÍACA MÁX ANGINA x EJERCICIO
## EDAD                  1.00000000  0.010307251       0.07064056      0.265638310  0.05557662      0.17719557    0.1
5301742       -0.36737055         0.18478532
## SEXO                  0.01030725  1.000000000       0.14393924     -0.006190966 -0.17030775      0.10983323   -0.0
2065464       -0.17072969         0.18312750
## TIPO DOLOR TORAX      0.07064056  0.143939237       1.00000000      0.035017478 -0.05318981      0.14023978    0.0
5055787       -0.26539514         0.21247464
## PRESIÓN ARTERIAL      0.26563831 -0.006190966       0.03501748      1.000000000  0.10091912      0.09287163    0.0
5902914       -0.09833084         0.13454092
## CORESTEROL            0.05557662 -0.170307752      -0.05318981      0.100919123  1.00000000     -0.05485501    0.0
4623150        0.07215950         0.04545846
## NIVEL DE AZÚCAR       0.17719557  0.109833233       0.14023978      0.092871629 -0.05485501      1.00000000    0.0
3366667       -0.12336361         0.05888944
## ECG EN REPOSO         0.15301742 -0.020654643       0.05055787      0.059029140  0.04623150      0.03366667    1.0
0000000        0.03680226         0.02065208
## FREC CARDÍACA MÁX    -0.36737055 -0.170729692      -0.26539514     -0.098330842  0.07215950     -0.12336361    0.0
3680226        1.00000000        -0.37732721
## ANGINA x EJERCICIO    0.18478532  0.183127496       0.21247464      0.134540924  0.04545846      0.05888944    0.0
2065208       -0.37732721         1.00000000
## OLDPEAK               0.24752994  0.095590850       0.08451111      0.176890078  0.06230374      0.03594124    0.0
7711587       -0.18634309         0.37165510
## PENDIENTE ST          0.16103760  0.036364404      -0.03765429      0.017546990  0.04571129      0.06589316    0.0
4976452       -0.04619899         0.19758205
## E. CARDIACA           0.15913980  0.144474318       0.40242504      0.053535720 -0.03116443      0.20237228    0.0
7378645       -0.20844073         0.27053481
##                          OLDPEAK PENDIENTE ST E. CARDIACA
## EDAD                  0.24752994   0.16103760  0.15913980
## SEXO                  0.09559085   0.03636440  0.14447432
## TIPO DOLOR TORAX      0.08451111  -0.03765429  0.40242504
## PRESIÓN ARTERIAL      0.17689008   0.01754699  0.05353572
## CORESTEROL            0.06230374   0.04571129 -0.03116443
## NIVEL DE AZÚCAR       0.03594124   0.06589316  0.20237228
## ECG EN REPOSO         0.07711587   0.04976452  0.07378645
## FREC CARDÍACA MÁX    -0.18634309  -0.04619899 -0.20844073
## ANGINA x EJERCICIO    0.37165510   0.19758205  0.27053481
## OLDPEAK               1.00000000   0.21639692  0.18230327
## PENDIENTE ST          0.21639692   1.00000000  0.44099197
## E. CARDIACA           0.18230327   0.44099197  1.00000000
```

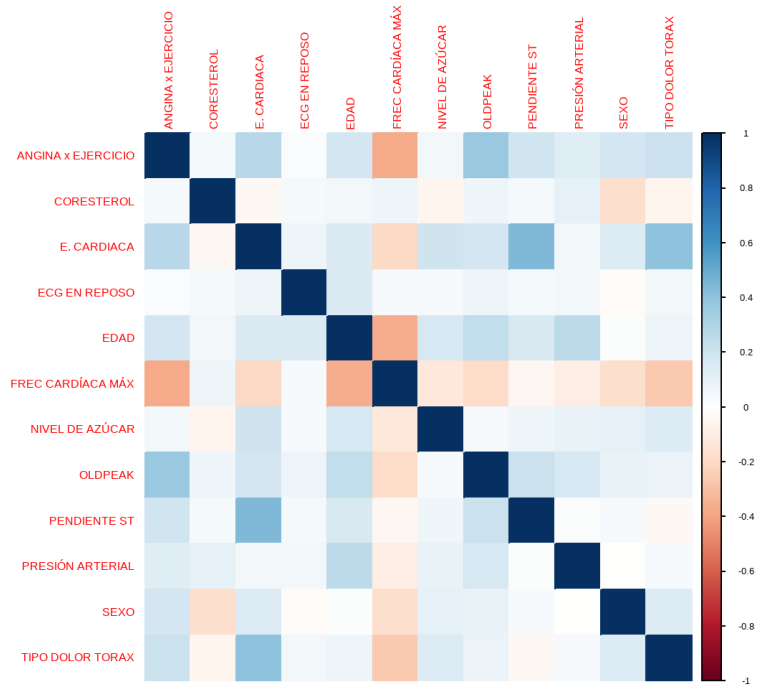
```
#Representación de las correlaciones
corrplot(cor_datos, method = "pie", type="upper")
```




```
#Representación de las correlaciones II
corrplot(cor_datos, method = 'shade', order = 'AOE')
```




```
#Representación de las correlaciones III
corrplot(cor_datos, method = 'color', order = 'alphabet')
```



Para representar las correlaciones, se ha usado diferentes métodos para ver las relaciones entre las características y verlo de una manera más clara.

2.4.5 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    PRESIÓN ARTERIAL          CORESTEROL     NIVEL DE AZÚCA
R       ECG EN REPOSO   FREC CARDÍACA MÁX
##          87.4315033           0.1808166           1.2233549         319.6802285        3071.0066876           0.167262
8           0.5577678         647.8786644
## ANGINA x EJERCICIO             OLDPEAK        PENDIENTE ST         E. CARDIACA
##           0.2369530           1.1930804           0.4791289           0.2475826
```

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

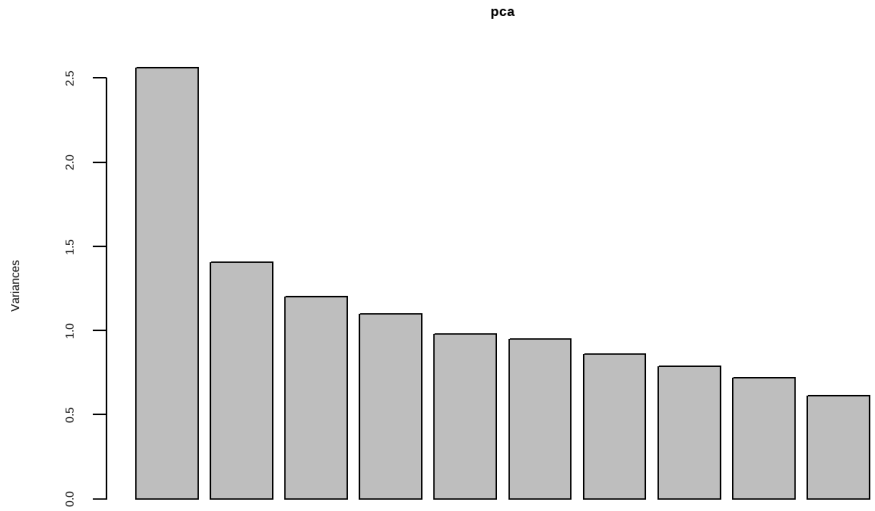
Lo siguiente es centrar y escalar las características, para que así las variables pierdan esa variabilidad. Una vez calculada la matriz se la asigno al pca

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

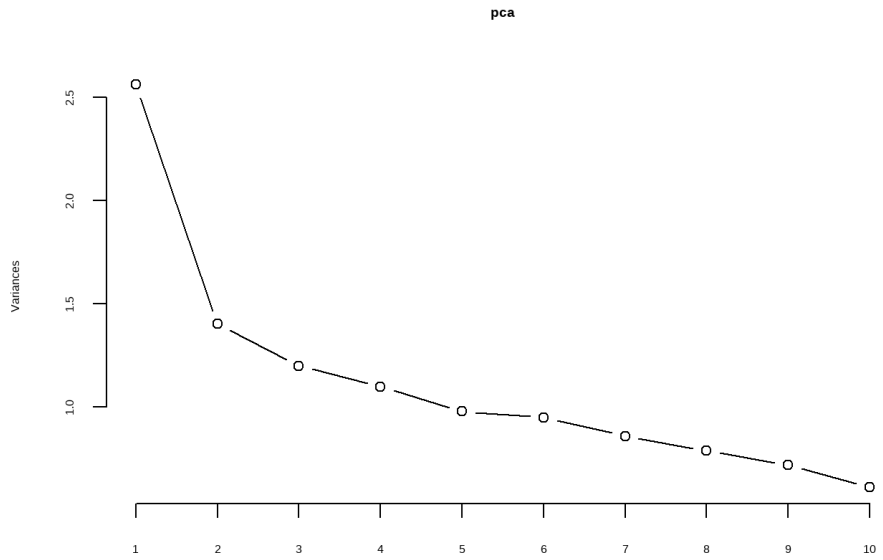
```
## Standard deviations (1, .., p=12):
##  [1] 1.6004587 1.1848672 1.0951841 1.0481039 0.9890992 0.9745326 0.9269463 0.8877397 0.8482690 0.7815714 0.67253
18 0.6153182
##
## Rotation (n x k) = (12 x 12):
##                                 PC1          PC2          PC3          PC4          PC5          PC6           PC7
PC8         PC9        PC10        PC11        PC12
## EDAD                0.34186886 -0.31597048  0.27284093 -0.19719939  0.22036848 -0.03054682  0.394699966 -0.08280
293  0.22183975  0.36136298 -0.52567759 -0.04105846
## SEXO                0.19695232  0.41874758  0.10159710  0.15624946  0.27521660  0.29068895 -0.504704790 -0.26613
424  0.47328322  0.20038552 -0.03160227  0.01325580
## TIPO DOLOR TORAX    0.29666462  0.35074626  0.05130859 -0.20673459 -0.62151831 -0.01899374 -0.054090760  0.21433
908 -0.07062946  0.25414392 -0.13918932  0.46811458
## PRESIÓN ARTERIAL    0.19450631 -0.38116848  0.34117489 -0.12963613  0.09326520 -0.14196344 -0.451908479  0.59104
748  0.20098241 -0.19698004  0.14233904  0.04249900
## CORESTEROL         -0.00374595 -0.49964402 -0.11013691  0.06643287 -0.48129712 -0.26512415 -0.280661867 -0.51650
117  0.25923726  0.10709899  0.08476148 -0.02138766
## NIVEL DE AZÚCAR     0.21033679  0.13463534  0.08129254 -0.55672296  0.28629720 -0.36105881 -0.243445587 -0.39721
389 -0.40491044 -0.14495703  0.05220850  0.06338703
## ECG EN REPOSO       0.09336401 -0.24452791 -0.10070730 -0.47737624 -0.12088456  0.77374750  0.032622175 -0.11710
842  0.01879421 -0.20963938  0.14467617  0.01221850
## FREC CARDÍACA MÁX  -0.38060216 -0.12436986 -0.37924919 -0.14428722  0.06495553  0.08338277 -0.443236654  0.16417
138 -0.17641796  0.15005294 -0.62169000 -0.01746038
## ANGINA x EJERCICIO  0.40752387 -0.01395628  0.04453332  0.40393678 -0.13276279  0.08562358 -0.068550323 -0.12478
324 -0.18542885 -0.62077660 -0.45065005 -0.02282787
## OLDPEAK             0.34104005 -0.25190114 -0.02266335  0.34168704  0.10252245  0.21743906 -0.169711382  0.02498
367 -0.57518714  0.48245931  0.22445107 -0.04209029
## PENDIENTE ST        0.27131324 -0.14004149 -0.64813472  0.06883012  0.31757589 -0.11948995  0.115175138  0.06047
606  0.19934936 -0.08228308  0.10069380  0.54153156
## E. CARDIACA         0.40579248  0.17380336 -0.44823765 -0.17158263 -0.13882935 -0.13737285  0.003426377  0.20485
733  0.11751791  0.04498669  0.04568829 -0.69045089
```

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

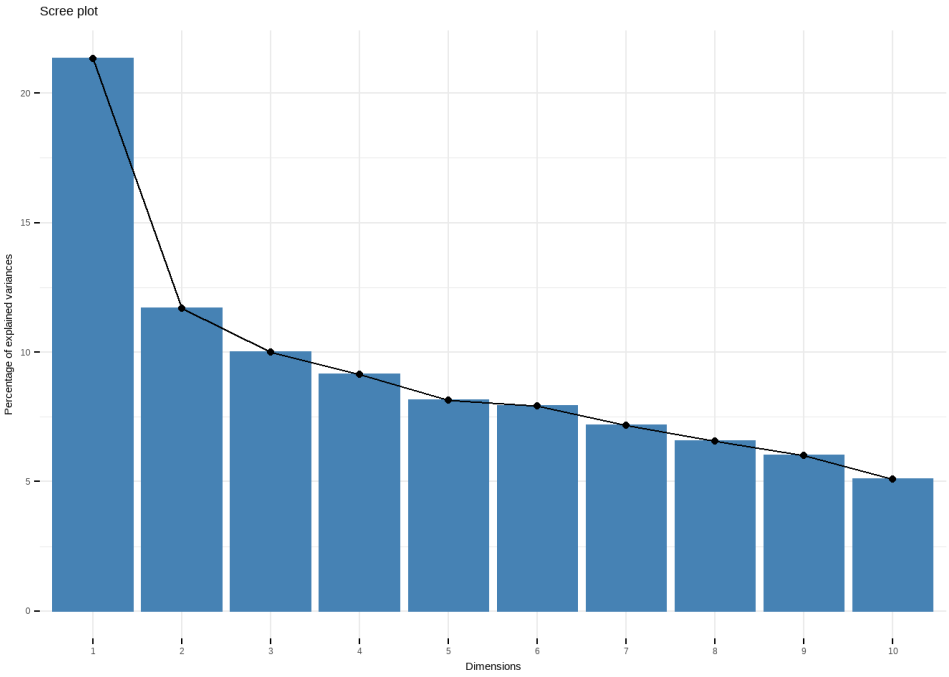
```
#Representación PCA´s anteriores
screeplot(pca)
```



```
plot(pca, type = "l")
```




```
#Juntamos las dos gráficas anteriores
fviz_eig(pca)
```



Como se ha dicho 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

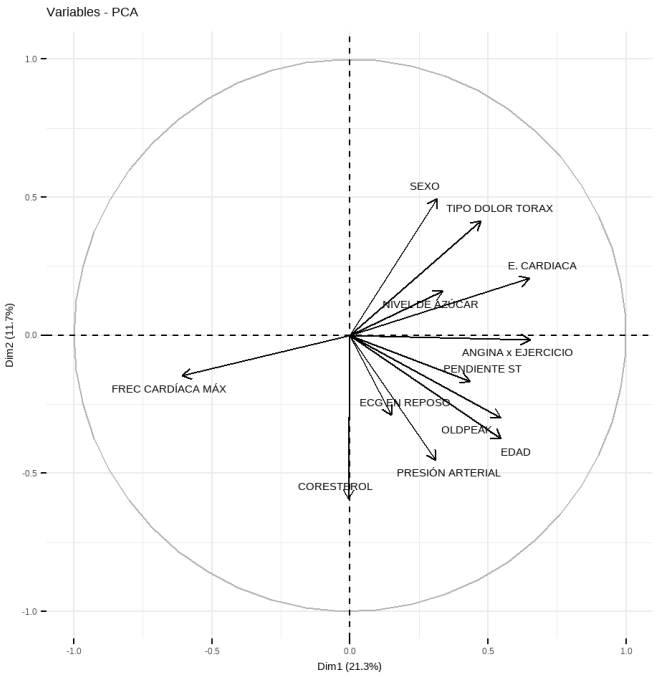
```
#Estadísticas de las componentes
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6    PC7     PC8     PC9    PC10    PC11    PC
12
## Standard deviation     1.6005 1.1849 1.09518 1.04810 0.98910 0.97453 0.9269 0.88774 0.84827 0.7816 0.67253 0.615
32
## Proportion of Variance 0.2135 0.1170 0.09995 0.09154 0.08153 0.07914 0.0716 0.06567 0.05996 0.0509 0.03769 0.031
55
## Cumulative Proportion  0.2135 0.3305 0.43040 0.52194 0.60347 0.68261 0.7542 0.81989 0.87985 0.9308 0.96845 1.000
00
```

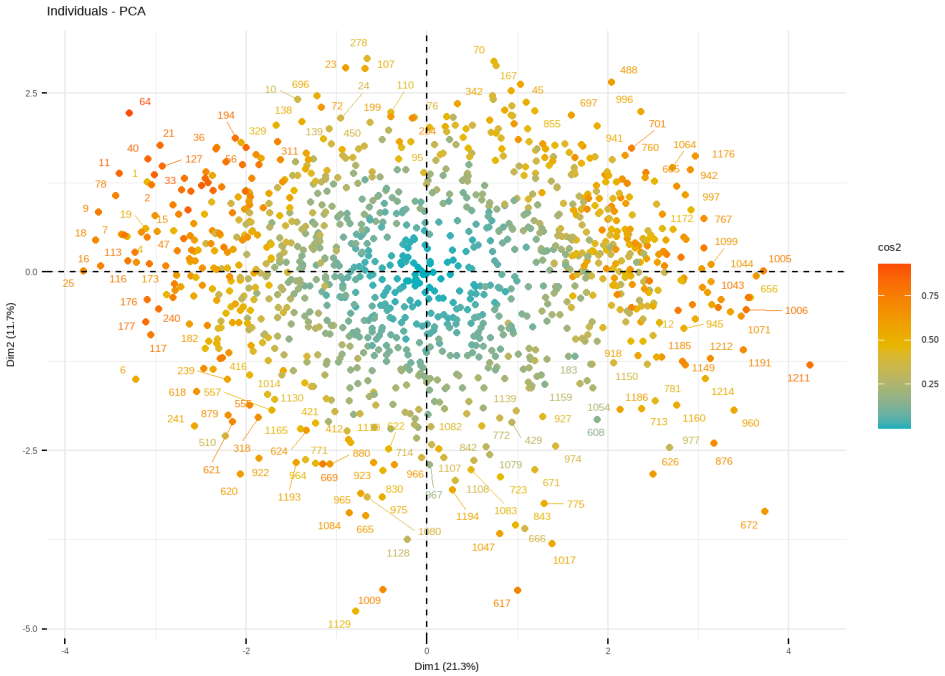
Viendo las estadísticas vemos que con las dos primeras componentes solamente podríamos explicar un 33,05% de los datos.Como no queremos perder información en el modelo, nos tendríamos que quedar con todas las componentes.

Para verlo de una manera visual, se va a representar la PCA de una manera gráfica.

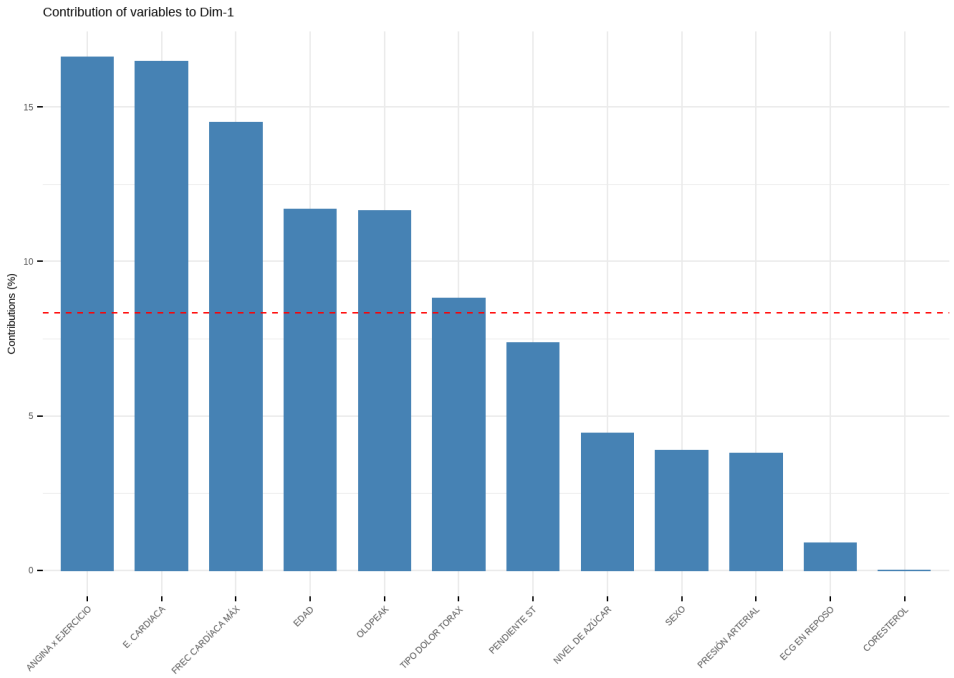
```
#Representación de variables sobre componentes principales
fviz_pca_var(pca, repel = TRUE, scale = 0)
```




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



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

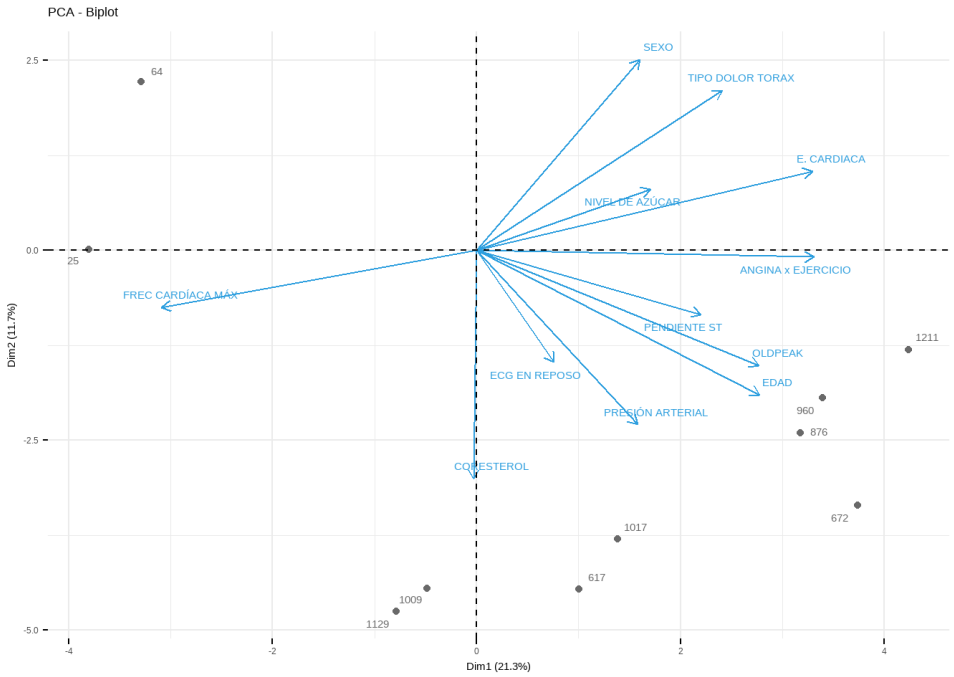


Una vez que hemos representada las variables y los individuos, se va a fusionar estas dos gráficas

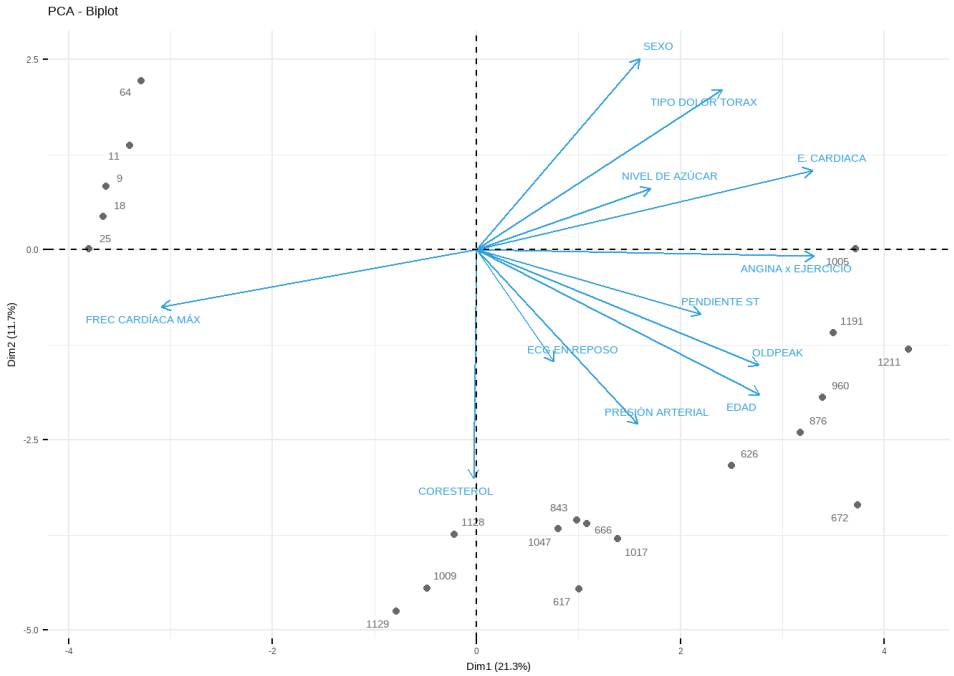
```
#Representación de variables y los individuos en la misma gráfica
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF", col.ind = "#696969")
```

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, 20 y 30 casos más influyentes

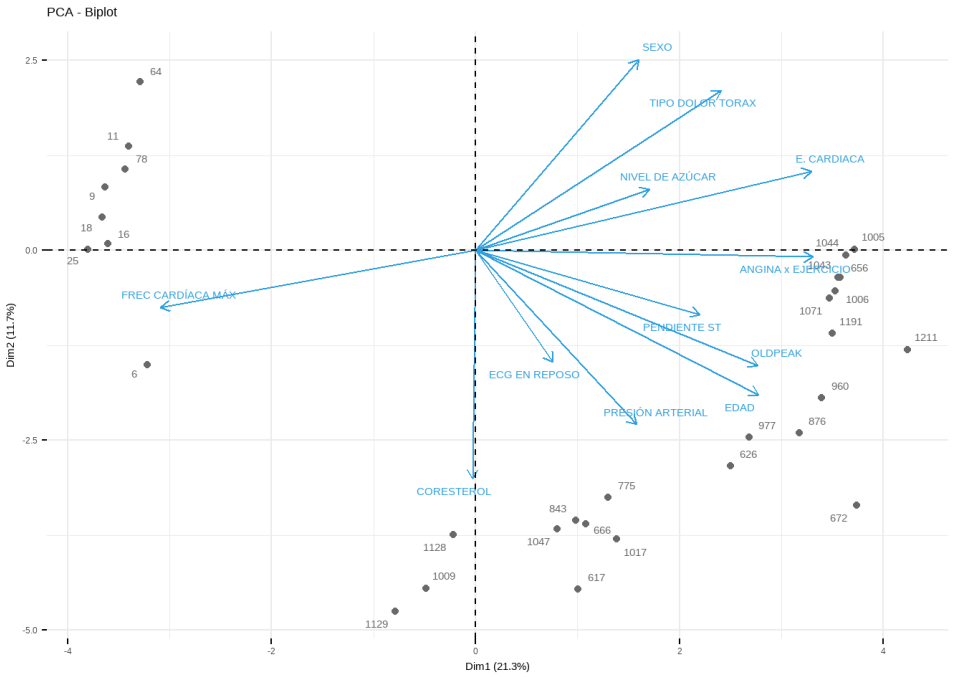
```
#Representación de variables y los 10 individuos más influyentes en la misma gráfica
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF", col.ind = "#696969", select.ind = list(contrib = 10))
```




```
#Representación de variables y los 10 individuos más influyentes en la misma gráfica
fviz_pca_biplot(pca, repel = TRUE, col.var = "#2E9FDF", col.ind = "#696969", select.ind = list(contrib = 20))
```




```
#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 = 30))
```



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.

3 Conclusiones

Una vez realizado todos los procedimientos anteriores, se ha obtenido un conjunto de datos ya preparados poder aplicar algoritmos (de clustering, regresión o clasificación) sin perder nunca de vista el objetivo que se pretende alcanzar en este estudio.

---

4 Criterios de evaluación

- 10%. Justificación de la elección del juego de datos donde se detalle el potencial analítico que se intuye.
- 10% Se plantea un problema propio de minería de datos, se detallan los objetivos analíticos y se explica detalladamente el procedimiento para darles solución.
- 20%. Información extraída del análisis exploratorio. Distribuciones, correlaciones, anomalías ...
- 20%. Explicación clara de cualquier tarea de limpieza, acondicionado o transformación que se realiza sobre los datos, justificando el motivo y mencionando las ventajas de la acción tomada.
- 20%. Se realiza un proceso de PCA o SVD donde se aprecia mediante explicaciones y comentarios que el estudiante entiende todos los pasos y se comenta extensamente el resultado final obtenido. Se usan las componentes obtenidas para apoyar el análisis propuesto.
- 20%. Se obtiene un conjunto de datos preparado y descrito adecuadamente, especificando las variables que se usarán en los procesos de modelado posterior, y como se aplicarán los métodos propuestos sobre ellas.