Tipología y ciclo de vida de los datos

Autor: Francisco Jose Ramirez Vicente

Mayo 2022

Contents

```
# https://cran.r-project.org/web/packages/ggplot2/index.html
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
# https://cran.r-project.org/web/packages/dplyr/index.html
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
```

```
#install.packages("hms")
#install.packages("zip")
#install.packages('car')
#install.packages("zip")
#install.packages("caTools")
#install.packages("ggcorrplot")
#install.packages("rpart.plot")
```

#PREGUNTA 1 ********** Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El dataset elegido es: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset El cual viene de la siguiente fuente de datos: https://archive.ics.uci.edu/ml/datasets/heart+disease

Este dataset es importante ya que nos permite, basándonos en una muestra previa, detectar posibles enfermedades caridiacas partiendo de una serie de atributos iniciales. De este modo si detectamos en un nuevo paciente estos mismos síntomas podríamos detectar algún tipo de enfermedad del corazón. La pregunta que pretende responder es si un determinado paciente, con unos atributos concretos, tiene o no tiene predisposición a tener una enfermedad cardíaca basándose en los parámetros de entrada.

#PREGUNTA 2 ********** Integración y selección de los datos de interés a analizar. Puede ser el resultado de adicionar diferentes datasets o una subselección útil de los datos originales, en base al objetivo que se quiera conseguir.

Vamos a analizar todos los atributos de entrada para de esta forma, entenderlos bien y poder realizar las operaciones posteriores de análisis:

Cargamos el dataset y obtenemos el número de files y luego la estructura de los datos para una primera referencia:

```
DataHeart <- read.csv('heart.csv',stringsAsFactors = FALSE)
filas=dim(DataHeart)[1]
filas
```

```
## [1] 303
```

```
str(DataHeart)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ 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 ...
```

En principio todo el dataset es interesante para realizar los análisis, aunque luego en la fase de análisis veremos cuáles son los más importantes para obtener resultados relevantes. Por otro lado, analizamos los diferentes atributos para entenderlos antes de comenzar con su análisis:

age Valor número y entero. Edad del paciente

sex Valor número y entero. Valor Binario, 1=male 0=female

cp Valor número y entero. CP significa Chest Pain (dolor de pecho) 0:angina típica, 1:angina atípica, 2:dolor no anginal y 3:asintomático.

trestbps Valor número y entero. Presión arterial en reposo (mmg/Hg)

chol Valor número y entero. Colesterol sérico (mg/dl)

fbs Valor número y entero. Aumento de azúcar en sangre, valor binario, si es mayor de 120 mg/dl, entonces 1=true 0=false

restecg Valor número y entero. Resultados de electrocardiogramas en reposo. Valores, 0:normal, 1:anomalía ST-T y 2:probable hipertrofía ventricular.

thalac Valor número y entero. Máximo número de pulsaciones en bmp.

exang Valor número y entero. Angina inducida por ejercicio.

oldpeak Valor número y decimal. Depresión del ST inducida por ejercicio en relación con el descanso.

slope Valor número y entero. Pendiente del segmento ST (pico de ejercicio). Puede tener los valores 1:pendiente ascendente, 2:plano y 3:pendiente descendente

ca Valor número y entero. número de vasos principales (de 0 a 3) que se han coloreado por fluoroscopia.

thal Valor número y entero. Tenemos los valores 3:normal, 6:defecto y 7:defecto reversible.

target Valor número y entero. Binario, se utiliza para la predicción.

PREGUNTA 3 ********** Limpieza de los datos

Como paso preliminar, vamos a modificar las columnas con nombres que podamos identificar de forma más amigable:

```
names(DataHeart)[names(DataHeart) == "age"] <- "Edad_Paciente"
names(DataHeart)[names(DataHeart) == "sex"] <- "Genero"
names(DataHeart)[names(DataHeart) == "cp"] <- "Dolor_Pecho"
names(DataHeart)[names(DataHeart) == "trestbps"] <- "Presion_Arterial_Reposo"
names(DataHeart)[names(DataHeart) == "chol"] <- "Colesterol_Serico"
names(DataHeart)[names(DataHeart) == "fbs"] <- "Aumento_azucar_sangre"
names(DataHeart)[names(DataHeart) == "restecg"] <- "ECG_Descanso"
names(DataHeart)[names(DataHeart) == "thalach"] <- "Pulsaciones_max_corazon_bpm"
names(DataHeart)[names(DataHeart) == "exang"] <- "Angina_inducida_ejercicio"
names(DataHeart)[names(DataHeart) == "oldpeak"] <- "Depresion_ST_ejercicio"
names(DataHeart)[names(DataHeart) == "slope"] <- "Pico_ejercicio_ST"
names(DataHeart)[names(DataHeart) == "ca"] <- "Num_vasos_floururo"
names(DataHeart)[names(DataHeart) == "thal"] <- "Thalassemia"
names(DataHeart)[names(DataHeart) == "target"] <- "Prediccion_Diagnostico"
```

Por otro lado, hay varios valores que se tratan como enteros (int) en vez de factores. Por ejemplo, los campos "Dolor_Pecho", "Aumento_azucar_sangre", "ECG_Descanso", "Num_vasos_floururo", "Thalassemia", "Prediccion_Diagnostico" y "Genero" deberían ser tratados como factores en vez de enteros. Para realizar dicha conversión, podemos usar el siguiente código:

```
# Primero almacenamos los datos originales en otro dataset
nuevoDataHeart <- DataHeart %>%
  # Comenzamos con la conversión de los datos de int a factor
  mutate(Genero = if_else(Genero == 1, "hombre", "mujer"),
         Angina_inducida_ejercicio = if_else(Angina_inducida_ejercicio == 1, "si" ,"no"),
         Pico_ejercicio_ST = as.factor(Pico_ejercicio_ST),
         Num_vasos_floururo = as.factor(Num_vasos_floururo),
         Thalassemia = as.factor(Thalassemia),
         ECG_Descanso=if_else(ECG_Descanso == 0, "normal",
                              if_else(ECG_Descanso == 1, "anomalia", "probable hipertrofia")),
         Dolor_Pecho = if_else(Dolor_Pecho == 0, "angina tipica",
                               if_else(Dolor_Pecho == 1, "angina atipica",
                                       if_else(Dolor_Pecho==2,"dolor no anginal","asintomatico"))),
         Pico_ejercicio_ST = as.factor(Pico_ejercicio_ST),
         Num_vasos_floururo = as.factor(Num_vasos_floururo),
         Thalassemia = as.factor(Thalassemia),
         Prediccion_Diagnostico = if_else(Prediccion_Diagnostico == 1, "SI", "NO")
         ) %>%
  mutate_if(is.character, as.factor) %>%
  dplyr::select(Genero,Angina_inducida_ejercicio,Pico_ejercicio_ST,Num_vasos_floururo,Thalassemia,ECG_D
```

Comprobamos que los cambios se han aplicado correctamente comparando los dos datasets:

```
head(DataHeart)
```

```
##   Edad_Paciente Genero Dolor_Pecho Presion_Arterial_Reposo Colesterol_Serico
```

```
## 1            63      1           3                     145              233
## 2            37      1           2                     130              250
## 3            41      0           1                     130              204
## 4            56      1           1                     120              236
## 5            57      0           0                     120              354
## 6            57      1           0                     140              192
##   Aumento_azucar_sangre ECG_Descanso Pulsaciones_max_corazon_bpm
## 1                     1           0                        150
## 2                     0           1                        187
## 3                     0           0                        172
## 4                     0           1                        178
## 5                     0           1                        163
## 6                     0           1                        148
##   Angina_inducida_ejercicio Depresion_ST_ejercicio Pico_ejercicio_ST
## 1                         0                   2.3                 0
## 2                         0                   3.5                 0
## 3                         0                   1.4                 2
## 4                         0                   0.8                 2
## 5                         1                   0.6                 2
## 6                         0                   0.4                 1
##   Num_vasos_floururo Thalassemia Prediccion_Diagnostico
## 1                  0           1                      1
## 2                  0           2                      1
## 3                  0           2                      1
## 4                  0           2                      1
## 5                  0           2                      1
## 6                  0           1                      1
```

```
head(nuevoDataHeart)
```

```
##   Genero Angina_inducida_ejercicio Pico_ejercicio_ST Num_vasos_floururo
## 1 hombre                        no                 0                  0
## 2 hombre                        no                 0                  0
## 3  mujer                        no                 2                  0
## 4 hombre                        no                 2                  0
## 5  mujer                        si                 2                  0
## 6 hombre                        no                 1                  0
##   Thalassemia ECG_Descanso      Dolor_Pecho Prediccion_Diagnostico
## 1           1       normal     asintomatico                    SI
## 2           2     anomalia dolor no anginal                    SI
## 3           2       normal   angina atipica                    SI
## 4           2     anomalia   angina atipica                    SI
## 5           2     anomalia    angina tipica                    SI
## 6           1     anomalia    angina tipica                    SI
##   Edad_Paciente Presion_Arterial_Reposo Colesterol_Serico Aumento_azucar_sangre
## 1            63                     145              233                     1
## 2            37                     130              250                     0
## 3            41                     130              204                     0
## 4            56                     120              236                     0
## 5            57                     120              354                     0
## 6            57                     140              192                     0
##   Pulsaciones_max_corazon_bpm Depresion_ST_ejercicio
## 1                         150                   2.3
## 2                         187                   3.5
```

```
## 3                          172                   1.4
## 4                          178                   0.8
## 5                          163                   0.6
## 6                          148                   0.4
```

3.1. ¿Los datos contienen ceros o elementos vacíos? Para comprobarlo realizamos el siguiente análsis de los datos:

```
colSums(is.na(nuevoDataHeart))
```

```
##                       Genero   Angina_inducida_ejercicio
##                            0                           0
##           Pico_ejercicio_ST          Num_vasos_floururo
##                            0                           0
##                  Thalassemia                ECG_Descanso
##                            0                           0
##                 Dolor_Pecho      Prediccion_Diagnostico
##                            0                           0
##               Edad_Paciente     Presion_Arterial_Reposo
##                            0                           0
##           Colesterol_Serico       Aumento_azucar_sangre
##                            0                           0
## Pulsaciones_max_corazon_bpm      Depresion_ST_ejercicio
##                            0                           0
```

```
colSums(nuevoDataHeart=="")
```

```
##                       Genero   Angina_inducida_ejercicio
##                            0                           0
##           Pico_ejercicio_ST          Num_vasos_floururo
##                            0                           0
##                  Thalassemia                ECG_Descanso
##                            0                           0
##                 Dolor_Pecho      Prediccion_Diagnostico
##                            0                           0
##               Edad_Paciente     Presion_Arterial_Reposo
##                            0                           0
##           Colesterol_Serico       Aumento_azucar_sangre
##                            0                           0
## Pulsaciones_max_corazon_bpm      Depresion_ST_ejercicio
##                            0                           0
```

Vemos que no hay elementos vacios ni nulos en el dataset

3.1. Identifica y gestiona los valores extremos

En primer lugar vemos con el comando summary un resumen de todos los valores, con sus máximos, mínimos, etc:

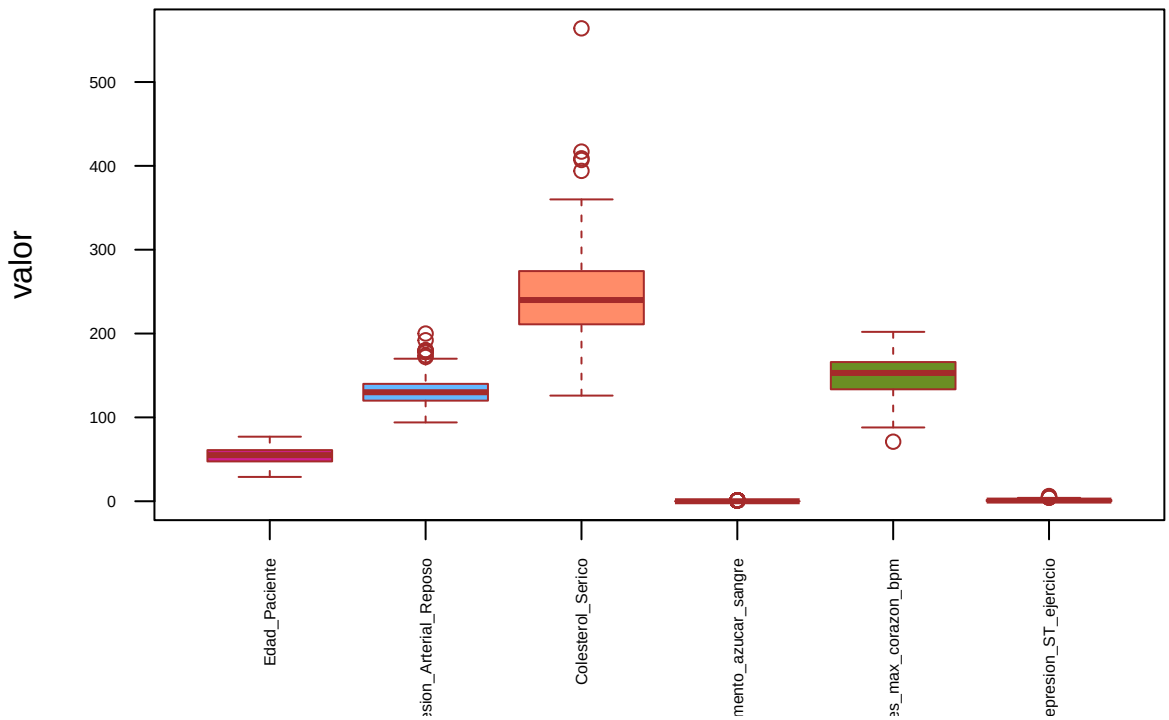
```
summary(nuevoDataHeart)
```

```
##     Genero    Angina_inducida_ejercicio Pico_ejercicio_ST Num_vasos_floururo
##  hombre:207   no:204                    0: 21             0:175
```

```
##  mujer : 96   si: 99                    1:140             1: 65
##                                         2:142             2: 38
##                                                           3: 20
##                                                           4:  5
##
##  Thalassemia               ECG_Descanso           Dolor_Pecho
##  0:  2       anomalia            :152   angina atipica  : 50
##  1: 18       normal              :147   angina tipica   :143
##  2:166       probable hipertrofia:  4   asintomatico    : 23
##  3:117                                  dolor no anginal: 87
##
##
##  Prediccion_Diagnostico Edad_Paciente   Presion_Arterial_Reposo
##  NO:138                 Min.   :29.00   Min.   : 94.0
##  SI:165                 1st Qu.:47.50   1st Qu.:120.0
##                         Median :55.00   Median :130.0
##                         Mean   :54.37   Mean   :131.6
##                         3rd Qu.:61.00   3rd Qu.:140.0
##                         Max.   :77.00   Max.   :200.0
##  Colesterol_Serico Aumento_azucar_sangre Pulsaciones_max_corazon_bpm
##  Min.   :126.0     Min.   :0.0000        Min.   : 71.0
##  1st Qu.:211.0     1st Qu.:0.0000        1st Qu.:133.5
##  Median :240.0     Median :0.0000        Median :153.0
##  Mean   :246.3     Mean   :0.1485        Mean   :149.6
##  3rd Qu.:274.5     3rd Qu.:0.0000        3rd Qu.:166.0
##  Max.   :564.0     Max.   :1.0000        Max.   :202.0
##  Depresion_ST_ejercicio
##  Min.   :0.00
##  1st Qu.:0.00
##  Median :0.80
##  Mean   :1.04
##  3rd Qu.:1.60
##  Max.   :6.20
```

Podemos identificar los outliers y los valores extremos de una manera gráfica usando bloxplot:

```
valores_extremos <- boxplot(nuevoDataHeart %>% select_if(is.numeric), main = 'Datos números cos y posible
```



Podemos identificar a simple vista varios valores outliers (círculos rojos) y para mostrarlos podemos visualizar los datos:

```
valores_extremos
```

```
## $stats
##      [,1] [,2]  [,3] [,4]  [,5] [,6]
## [1,] 29.0   94 126.0    0  88.0  0.0
## [2,] 47.5  120 211.0    0 133.5  0.0
## [3,] 55.0  130 240.0    0 153.0  0.8
## [4,] 61.0  140 274.5    0 166.0  1.6
## [5,] 77.0  170 360.0    0 202.0  4.0
##
## $n
## [1] 303 303 303 303 303 303
##
## $conf
##          [,1]     [,2]     [,3] [,4]   [,5]      [,6]
## [1,] 53.77462 128.1846 234.2362    0 150.05 0.6547702
## [2,] 56.22538 131.8154 245.7638    0 155.95 0.9452298
##
## $out
##  [1] 172.0 178.0 180.0 180.0 200.0 174.0 192.0 178.0 180.0 417.0 564.0 394.0
## [13] 407.0 409.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
## [25]   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
## [37]   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
## [49]   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0  71.0
```

```
## [61]   4.2   6.2   5.6   4.2   4.4
##
## $group
##  [1] 2 2 2 2 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [39] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 6 6 6 6 6
##
## $names
## [1] "Edad_Paciente"               "Presion_Arterial_Reposo"
## [3] "Colesterol_Serico"           "Aumento_azucar_sangre"
## [5] "Pulsaciones_max_corazon_bpm" "Depresion_ST_ejercicio"
```

Los valores extremos los podemos encontrar en $out y podemos confirmar 20: 172.0 178.0 180.0 180.0 200.0 174.0 192.0 178.0 180.0 417.0 564.0 394.0 407.0 409.0 71.0 4.2 6.2 5.6 4.2 4.4

Estos valores los podríamos sustituir por ejemplo los máximos con la mediana y los mínimos con la media. En este caso no aplicaré estos cambios por no tener un efecto significativo en el resto del análisis.

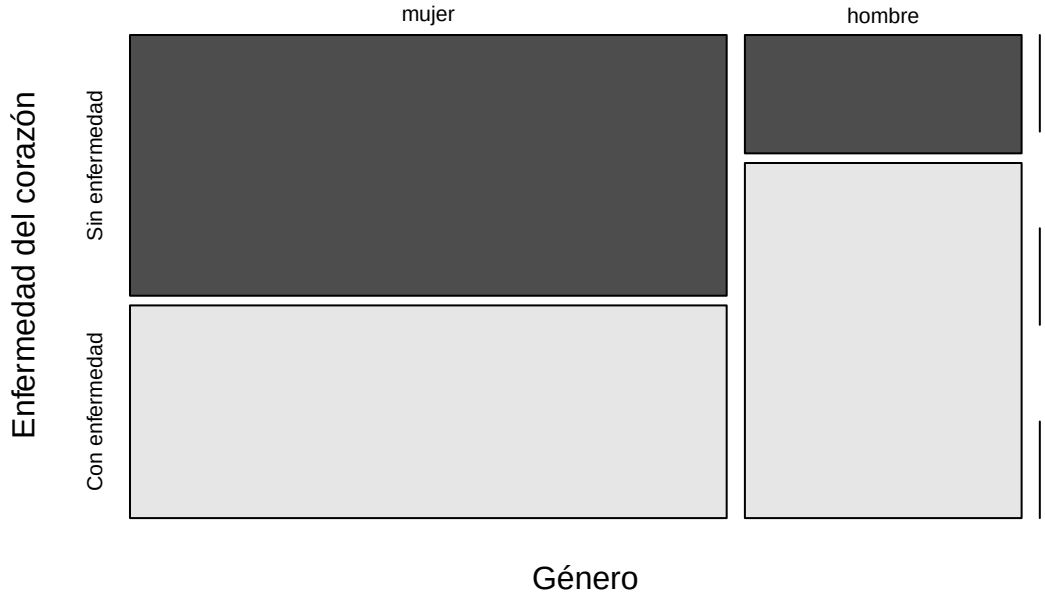
PREGUNTA 4 ********** Análisis de los datos

4.1. Selección de los grupos de datos que se quieren analizar/comparar (p. e., si se van a comparar grupos de datos, ¿cuáles son estos grupos y qué tipo de análisis se van a aplicar?)

El primer grupo de datos que podríamos comparar es por género y ver cuáles tienen diagnosticados una enfermedad de corazón y cuantos no:

```
heart = nuevoDataHeart
levels(heart$Prediccion_Diagnostico) = c("Sin enfermedad","Con enfermedad")
levels(heart$Genero) = c("mujer","hombre","")
mosaicplot(heart$Genero ~ heart$Prediccion_Diagnostico, col = subs_pal,
           main="Enfermedad por géneros", shade=FALSE,color=TRUE,
           xlab="Género", ylab="Enfermedad del corazón")
```

```
## Warning: In mosaicplot.default(table(mf), main = main, ...) :
##  extra argument 'col' will be disregarded
```

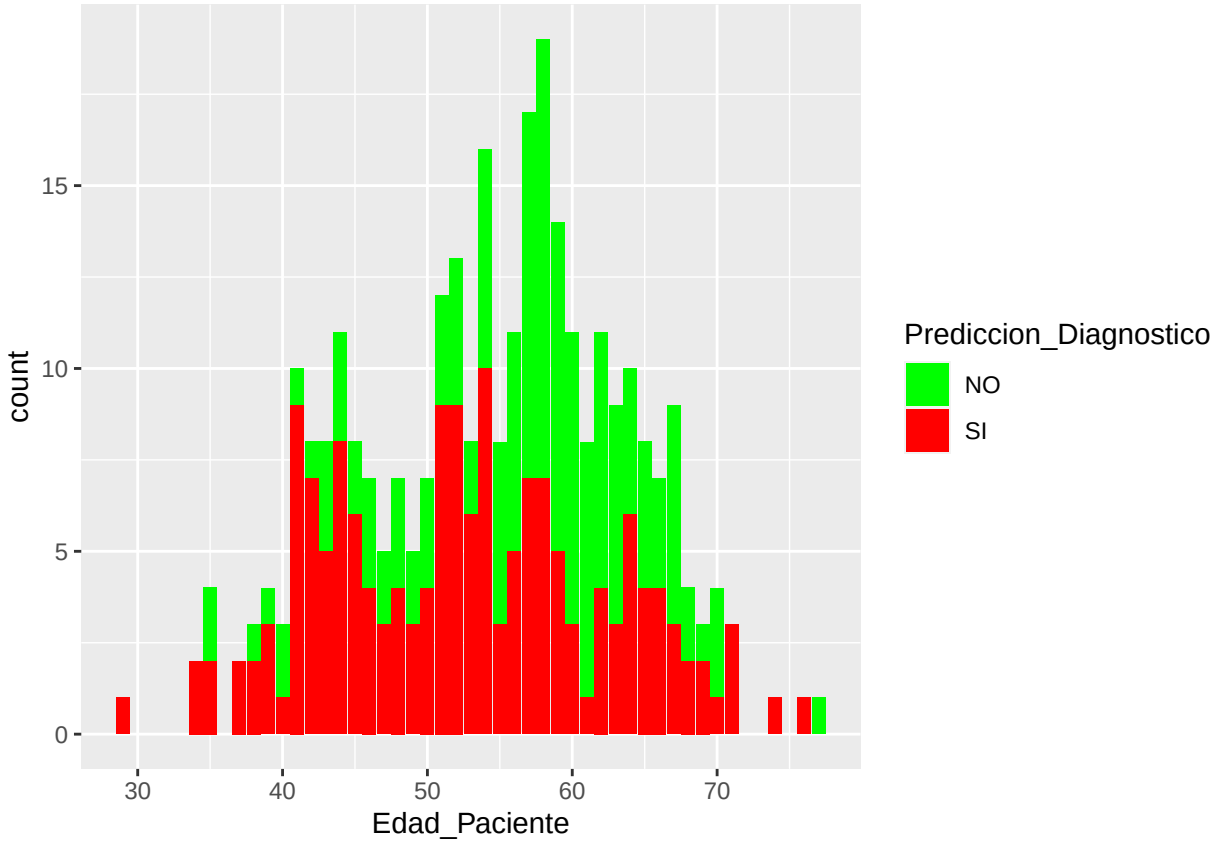


Destacar que por géneros, es el hombre el que tiene más enfermedades de corazón si tenemos en cuenta los parámetros del datset.

A partir de este punto podríamos elegir muchos grupos de datos. es decir, prácticamente cada característica de cada elemento del dataset se podría ir agrupando y comparando con el resto para sacar conlusiones de relación. Pero por restricciones en la entrega, me centraré sólo en uno más pero luego desarrollaré más análisis en el punto 4.3.

Por ejemplo, sin nos centramos en los dolores de pecho y la edad:

```
ggplot(nuevoDataHeart, aes(x=Edad_Paciente, fill=Prediccion_Diagnostico)) + geom_bar()+scale_fill_manual
```



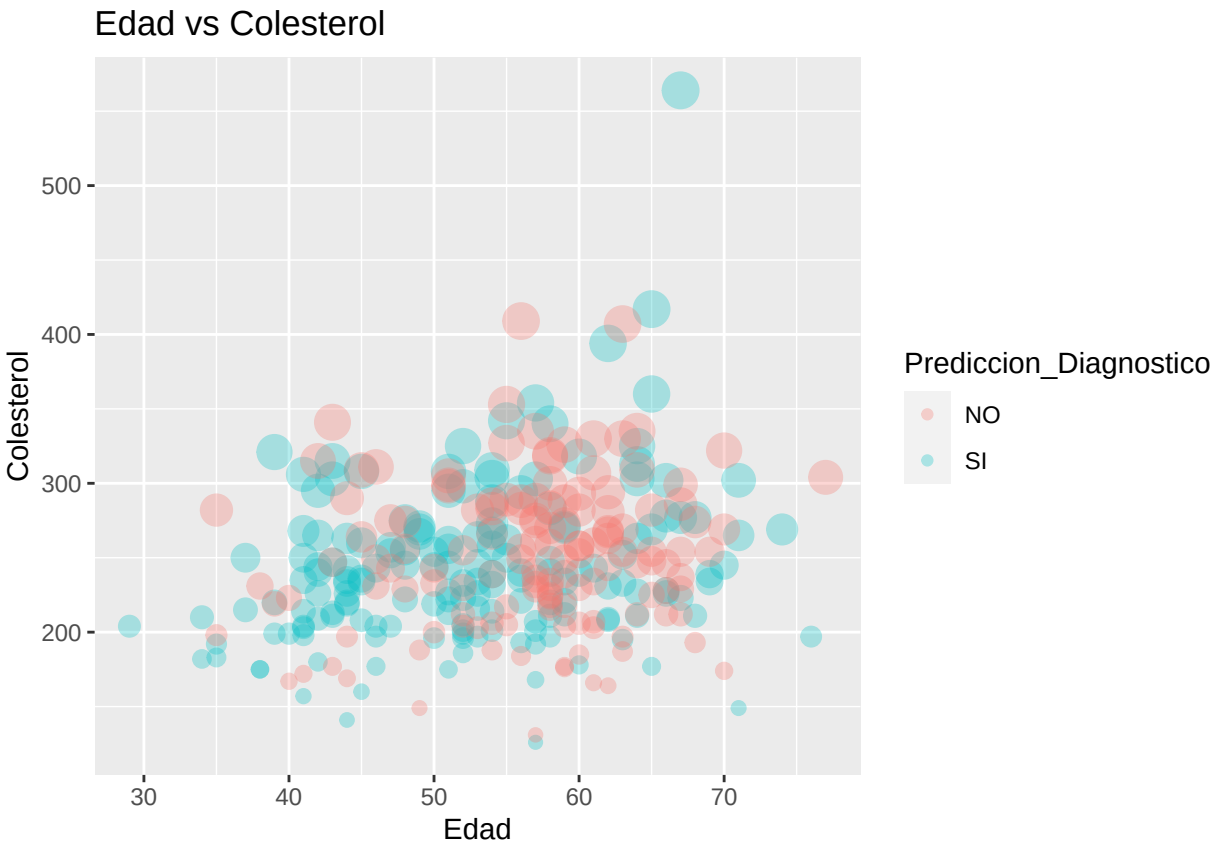
Podemos destacar de esta gráfica que la edad donde más casos se diagnostican está comprendida entre los 40 y 60 años aproximadamente (hombres y mujeres)

Ahora podríamos ir comparando diferentes grupos. Por ejemplo, podríamos comprobar los niveles de colesterol, con la edad y ver posibles distribuciones de casos positivos de enfermedad:

```
ggplot(nuevoDataHeart,aes(x = Edad_Paciente, y = Colesterol_Serico,color=Prediccion_Diagnostico, size =
    geom_point(alpha=0.3) + guides(size=FALSE) + xlab("Edad") + ylab("Colesterol") + ggtitle("Edad vs C
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
## Warning: Using size for a discrete variable is not advised.
```



Vemos que entre rangos de colesterol entre 200 y 350 y además entre una franja de edad de entre 50 y 65 años es cuando más casos de enferdad se diagnostican.

Estos son sólo tres ejemplos de todos los posibles agrupamientos que podrímos realizar.

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Comprobación de la normalidad:

Para comprobarlo comprobaré qué género tiene más posibilidades de diagnosticar una enfermedad de corazón (al igual que hice en el punto 4.1). Utilizaré el dataset original ya que tiene los valores numéricos.

Comenzamos por los hombres:

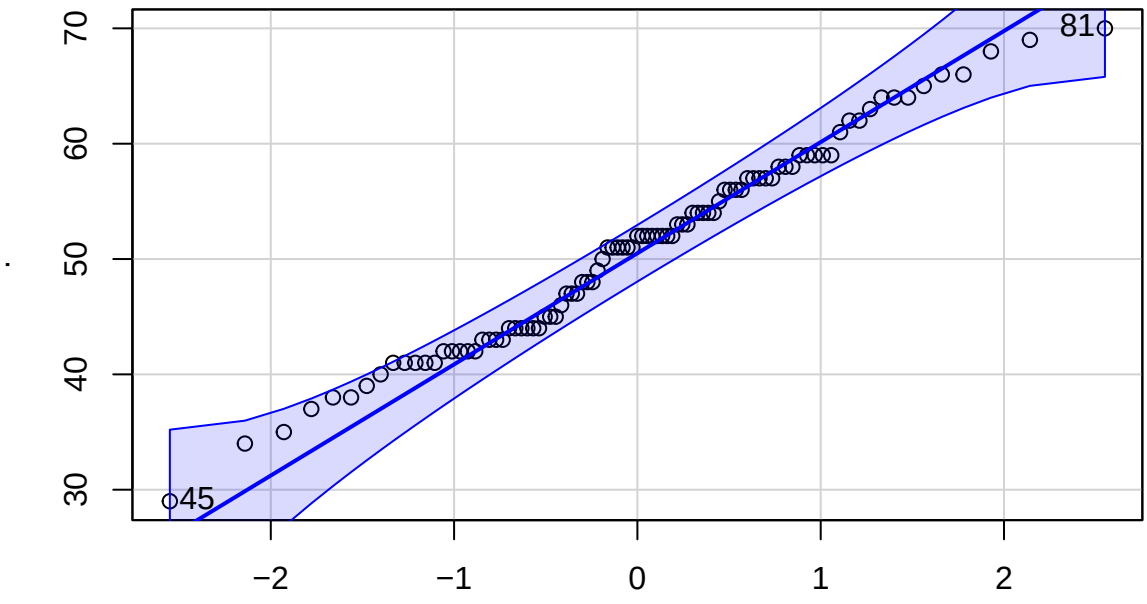
```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
DataHeart_hombres_con_enfermedad <- DataHeart %>% filter(Prediccion_Diagnostico == 1)
DataHeart_hombres_edad <- DataHeart_hombres_con_enfermedad %>% filter(Genero==1)
DataHeart_hombres_edad$Edad_Paciente %>% qqPlot(dist="norm", xlab = "Edad de los hombres con diagnóstic
```

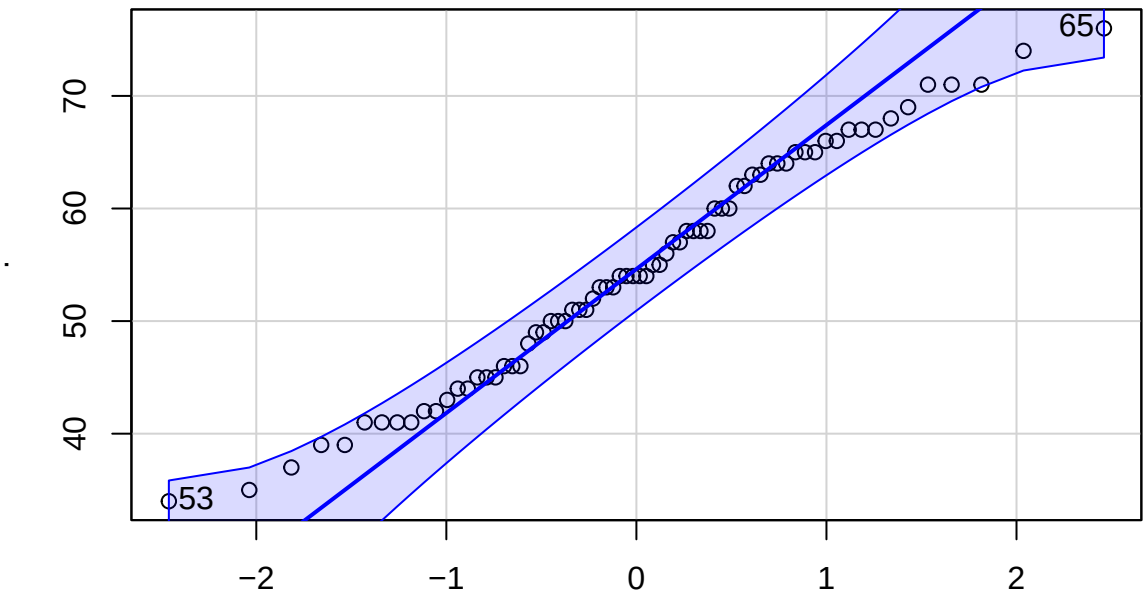


Edad de los hombres con diagnóstico de enfedad cardiaca

```
## [1] 45 81
```

Podemos ver que se encuentra repartida entre los 45 y los 81 años Vamos a ver ahora el de las mujeres:

```
library(car)
DataHeart_mujeres_con_enfermedad <- DataHeart %>% filter(Prediccion_Diagnostico == 1)
DataHeart_mujeres_edad <- DataHeart_mujeres_con_enfermedad %>% filter(Genero==0)
DataHeart_mujeres_edad$Edad_Paciente %>% qqPlot(dist="norm", xlab = "Edad de las mujeres con diagnóstic
```



Edad de las mujeres con diagnóstico de enfedad cardiaca

```
## [1] 65 53
```

En este caso vemos que está comprendida entre los 53 y los 65 años.

En ambos casos podemos observar que están dentro del rango de confianza asignado (zona azul) y teniendo en cuenta la población de ambos podemos asumir que es normalmente distribuida.

Comprobación de la homogeneida de la varianza: Utilizaré el método de Levene donde la variable1 será hombre y la variable2 será mujer.

```
#library(car)
leveneTest(Edad_Paciente ~ Genero, data = nuevoDataHeart)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1   0.363 0.5473
##       301
```

Observando los resultados podemos ver que la varianza entre la edad de los géneros tiene un valor similar.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Matriz Correlación

El primer método de análisis que voy a realizar es una Matriz de correlación, ya que nos dará una primera aproximación de posibles relaciones entre las catacterísticas y los diagnósticos de enfermedad. El primer paso será utilizar las variables no categóricas que tenemos:

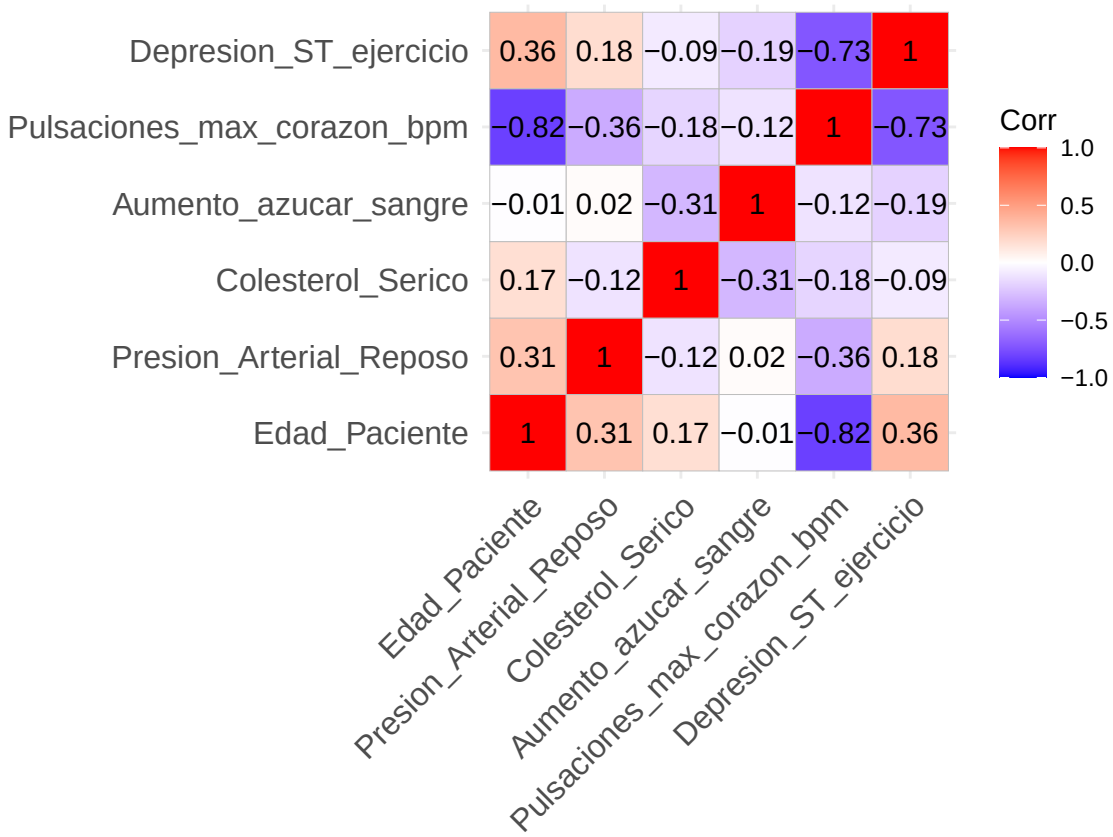
```
DataHeart_MatrixCorr <- cor(nuevoDataHeart[,9:14])
DataHeart_MatrixCorr
```

```
##                             Edad_Paciente Presion_Arterial_Reposo
## Edad_Paciente                   1.0000000              0.27935091
## Presion_Arterial_Reposo         0.2793509              1.00000000
## Colesterol_Serico               0.2136780              0.12317421
## Aumento_azucar_sangre           0.1213076              0.17753054
## Pulsaciones_max_corazon_bpm    -0.3985219             -0.04669773
## Depresion_ST_ejercicio          0.2100126              0.19321647
##                             Colesterol_Serico Aumento_azucar_sangre
## Edad_Paciente                     0.213677957           0.121307648
## Presion_Arterial_Reposo           0.123174207           0.177530542
## Colesterol_Serico                 1.000000000           0.013293602
## Aumento_azucar_sangre             0.013293602           1.000000000
## Pulsaciones_max_corazon_bpm      -0.009939839          -0.008567107
## Depresion_ST_ejercicio            0.053951920           0.005747223
##                             Pulsaciones_max_corazon_bpm Depresion_ST_ejercicio
## Edad_Paciente                            -0.398521938            0.210012567
## Presion_Arterial_Reposo                  -0.046697728            0.193216472
## Colesterol_Serico                        -0.009939839            0.053951920
## Aumento_azucar_sangre                    -0.008567107            0.005747223
## Pulsaciones_max_corazon_bpm               1.000000000           -0.344186948
## Depresion_ST_ejercicio                   -0.344186948            1.000000000
```

```
cor<-cor(DataHeart_MatrixCorr, method="pearson")
print(cor, digits= 1)
```

```
##                             Edad_Paciente Presion_Arterial_Reposo
## Edad_Paciente                        1.00                    0.31
## Presion_Arterial_Reposo              0.31                    1.00
## Colesterol_Serico                    0.17                   -0.12
## Aumento_azucar_sangre               -0.01                    0.02
## Pulsaciones_max_corazon_bpm         -0.82                   -0.36
## Depresion_ST_ejercicio               0.36                    0.18
##                             Colesterol_Serico Aumento_azucar_sangre
## Edad_Paciente                            0.17                -0.01
## Presion_Arterial_Reposo                 -0.12                 0.02
## Colesterol_Serico                        1.00                -0.31
## Aumento_azucar_sangre                   -0.31                 1.00
## Pulsaciones_max_corazon_bpm             -0.18                -0.12
## Depresion_ST_ejercicio                  -0.09                -0.19
##                             Pulsaciones_max_corazon_bpm Depresion_ST_ejercicio
## Edad_Paciente                                    -0.8                    0.36
## Presion_Arterial_Reposo                          -0.4                    0.18
## Colesterol_Serico                                -0.2                   -0.09
## Aumento_azucar_sangre                            -0.1                   -0.19
## Pulsaciones_max_corazon_bpm                       1.0                   -0.73
## Depresion_ST_ejercicio                           -0.7                    1.00
```

```
library(ggcorrplot)
ggcorrplot(cor,lab = T)
```



A simple vista, podemos ver relaciones de nivel positivo entre por ejemplo la presión arterial en reposo con la edad. Lo mismo ocurre con la edad y la depresión ST ejercicio, es decir esta aumenta con la edad.

Regresión logística

Para este proceso, primero dividimos el dataset para el testeo (test) y el entrenamiento (training). De esta forma podemos predecir qué pacientes pueden llegar a tener una enfermedad de corazón.

```
set.seed(100)
library(caTools)
dataset_heart_split=sample.split(nuevoDataHeart$Prediccion_Diagnostico, SplitRatio = 0.75)
```

Separamos en dos datasets para entrenamiento y para test:

```
Train=subset(nuevoDataHeart,dataset_heart_split == TRUE)
Test=subset(nuevoDataHeart,dataset_heart_split == FALSE)
```

Creamos el modelo:

```
Modelo_Heart<-glm(Prediccion_Diagnostico~.,data= Train,family = "binomial")
Train$Prediccion<-fitted(Modelo_Heart)
head(select(Train, Prediccion_Diagnostico, Prediccion))
```

```
##   Prediccion_Diagnostico Prediccion
## 1                     SI  0.9521181
## 2                     SI  0.7366887
```

```
## 3                    SI  0.9867965
## 4                    SI  0.9614477
## 5                    SI  0.9771716
## 6                    SI  0.6560157
```

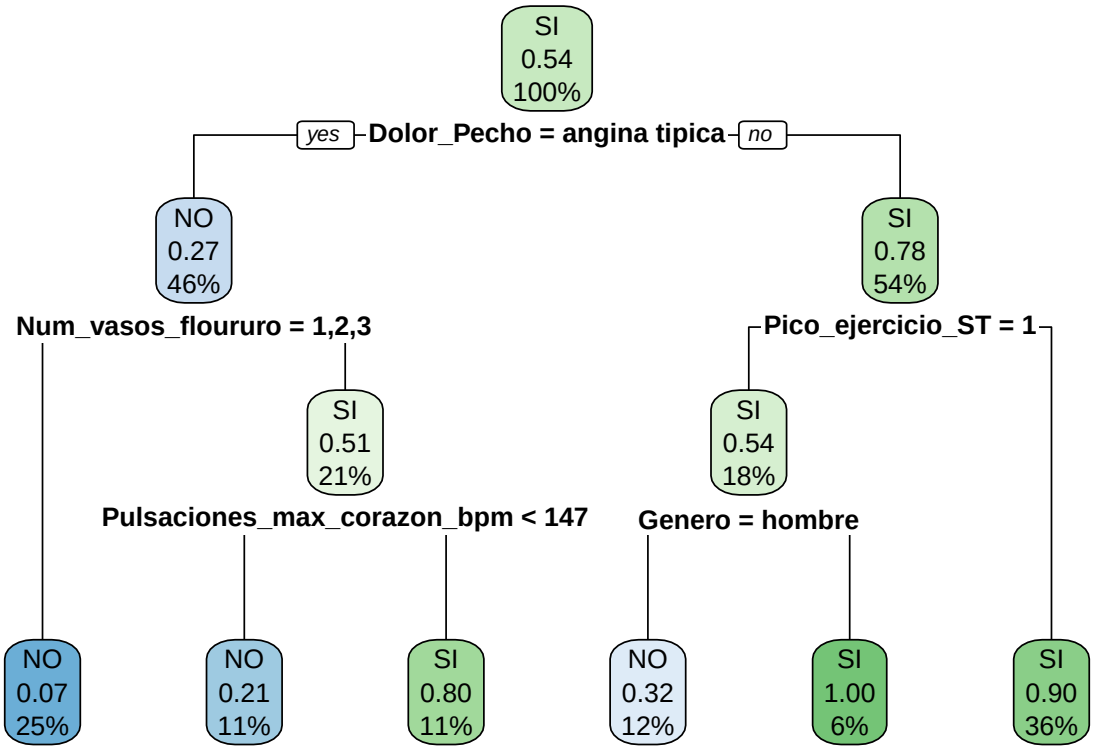
Nota: sólo muestro los primeros valores por limitación del documento. Comparando la columna "Preddiccion_Diagnostico" y la nueva que hemos añadido con la prediccion podemos comprobar que se ajusta a los datos del dataset y la predicción. Por ejemplo el paciente 1 tiene diagnosticada una enfermedad y la predicción que hemos realizado le asigna un 95% (0.95) de posibilidades de tenerla.

Arbol de decisión

Reutilizamos los datasets creados antes para Train y Test, inicializamos a NULL la columna de predicción:

```
#install.packages("rpart.plot")
library(rpart)
Train$Prediccion<-NULL
Test$Prediccion<-NULL
```

```
tree<-rpart(Prediccion_Diagnostico~.,method = "class",data = Train)
library(rpart.plot)
rpart.plot(tree)
```



En este árbol de ejemplo comenzamos por el nodo principal donde podemos ver la probabilidad de tener enfermedad mostrando la probabilidad total (0.54%). Luego preguntamos si Dolor_Pecho = angina tipica y así vamos recorriendo los diferentes nodos comprobando las probabilidades hasta llegar a los nodos finales donde aparecerá un posible diagónstico.

PREGUNTA 5 ********** Esta pregunta ha ido respondiendo entre los apartados anteriores

---

PREGUNTA 6 ********** ¿Cuáles son las conclusiones? El dataset heart disease contiene información muy útil a la hora de detectar posibles enfermedades de corazón. El principal problema al cual nos encotramos es a la normalización y el entendimiento de los conceptos médicos de cada característica del dataset. Una vez realizada esta fase de limpieza, los cálculos realizados permiten encontrar modelos con una suficiente garantía de diagnosticar los posibles casos fuera de este dataset.

En una primera fase de análisis (punto 4.1) hemos encontrado: * El género masculino es el que tiene más diagnósticos positivos de enferdad * La edad comprendida entre los 40 y los 60 años es la más propensa a tener un diagnostico positivo de enfermedad en ambos sexos. * Entre rangos de colesterol de 200 y 350 y además entre una franja de edad de entre 50 y 65 años es cuando más casos de enferdad se diagnostican.

En una segunda fase de análisis (predicción, punto 4.3) encontramos: * Matriz de correlación que nos muestra las relaciones entre los valores y tener o no enfermedad * La regresión lineal nos permite calcular nuevos casos. Los resultados con las muestras obtenidas son generalmente positivos (habría que hacer un análsis de falsos positivos y negativos más profundo). * Finalmente, el árbol de decisión nos permitirá obtener una predicción siguiendo el camino de los valores a estudiar.

¿Los resultados pueden responder al problema? Como hemos podido observar, en la parte de análisis encontramos herramientas suficientes para responder al problema para nuevos casos que se tengamos que analiza (4.3). Todos ellos aproximan con un alto porcentaje de acierto las opciones de diagnosticar un caso basándonos en las características de cada inviduo del dataset.

PREGUNTA 7 ********** Se publica el código fuente

PARTICIPANTES: **********

Contribuciones Firma * Investigación previa FJRV * Redacción de las respuestas FJRV * Desarrollo Código FJRV FJRV: Francisco José Ramírez Vicente