Práctica 2 - Limpieza y análisis de datos

Tipología y ciclo de vida de los datos
Máster Universitario en Ciencia de Datos
Universitat Oberta de Catalunya

Rafael López García y Carlos Luis Gento de Celis

Enero 2022

Índice

1 Descripción del dataset

1.1 Origen de los datos, agradecimientos y preguntas que se pretenden responder

El dataset elegido para esta práctica es Cleveland Heart Disease, del sitio web UCI Machine Learning Repository. Estos datos se pueden obtener a través de esta URL: *https://archive.ics.uci.edu/ml/datasets/Heart+Disease*. Los datos fueron recogidos por la Cleveland Clinic Foundation por Robert Detrano.

El dataset elegido contiene 14 atributos con información demográfica y médica de pacientes a los que se les ha detectado la presencia de enfermedad del corazón y de pacientes que estaban sanos. En ese sentido, este dataset contiene información valiosa que puede ser utilizada para detectar la presencia de enfermedad en el corazón a partir de los resultados de diferentes pruebas médicas. Por tanto, la pregunta que queremos intentar responder es qué características demográficas de los pacientes y sus resultados médicos nos pueden ser factores de riesgo o de protección frente a una enfermedad del corazón y, por tanto, podrían ayudar a detectar su presencia.

1.2 Descripción de las variables

El dataset que vamos a utilizar contiene las siguientes 14 variables:

age: Edad de la observación en años.

sex: Sexo de la observación.
- Valor 0: Mujer
- Valor 1: Hombre

cp: Tipo de dolor de pecho.
- Valor 1: típico angina
- Valor 2: atípico angina
- Valor 3: Sin dolor en angina
- Valor 4: asintomático

trestbps: presión arterial en reposo (en mm Hg al ingreso en el hospital)

chol: colesterol sérica en mg / dl

fbs: azúcar en sangre en ayunas > 120 ml/dl.
- Valor 0: No
- Valor 1: Sí

restecg: resultados electro en reposo.
- Valor 0: normal
- Valor 1: tener anomalías en la onda ST-T (inversiones de la onda T y / o elevación o depresión del ST> 0,05 mV)
- Valor 2: muestra hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes

thalach: frecuencia cardíaca máxima alcanzada durante la prueba de esfuerzo con Talio

exang: angina inducida por ejercicio.
- Valor 0: No
- Valor 1: Sí

oldpeak: Depresión del ST inducida por el ejercicio en relación con el reposo

slope: la pendiente del segmento ST de ejercicio pico
- Valor 1: Creciente
- Valor 2: Plana
- Valor 3: Decreciente

ca: número de vasos principales (0-3) coloreados por la fluoroscopia

thal: Resultado de la prueba de esfuerzo con Talio.
- Valor 3: normal
- Valor 6: defecto fijo
- Valor 7: defecto reversible

target: Diagnóstico de enferdad del corazón.
- Valor 0: No hay presencia riesgo de infarto
- Valores 1 a 4: Sí hay presencia de riesgo de infarto

2 Integración y selección de los datos de interés

En primer lugar cargamos las librerías que vamos a utilizar durante la práctica:

```
if (!require('knitr')) install.packages('knitr', dependencies=TRUE); library(knitr)
if (!require('mlr')) install.packages('mlr', dependencies=TRUE); library(mlr)
if (!require('tidyverse')) install.packages('tidyverse', dependencies=TRUE); library(tidyverse)
if (!require('GGally')) install.packages('GGally', dependencies=TRUE); library(GGally)
if (!require('cowplot')) install.packages('cowplot', dependencies=TRUE); library(cowplot)
if (!require('dplyr')) install.packages('dplyr', dependencies=TRUE); library(dplyr)
if (!require('tidyr')) install.packages('tidyr', dependencies=TRUE); library(tidyr)
if (!require('readr')) install.packages('readr', dependencies=TRUE); library(readr)
if (!require('magrittr')) install.packages('magrittr', dependencies=TRUE); library(magrittr)
if (!require('moments')) install.packages('moments', dependencies=TRUE); library(moments)
if (!require('modeest')) install.packages('modeest', dependencies=TRUE); library(modeest)
if (!require('factoextra')) install.packages('factoextra', dependencies=TRUE); library(factoextra)
if (!require('corrplot')) install.packages('corrplot', dependencies=TRUE); library(corrplot)
if (!require('caTools')) install.packages('caTools', dependencies=TRUE); library(caTools)
if (!require('VIM')) install.packages('VIM', dependencies=TRUE); library(VIM)
if (!require('ROCR')) install.packages('ROCR', dependencies=TRUE); library(ROCR)
if (!require('rpart')) install.packages('rpart', dependencies=TRUE); library(rpart)
if (!require('rpart.plot')) install.packages('rpart.plot', dependencies=TRUE); library(rpart.plot)
if (!require('caret')) install.packages('caret', dependencies=TRUE); library(caret)
if (!require('randomForest')) install.packages('randomForest', dependencies=TRUE); library(randomForest)
```

Cargamos el fichero de datos de cleveland y añadimos los nombres de las variables y comprobamos que se ha cargado correctamente:

```
df <- read.csv('data/processed.cleveland.data', na = "?", stringsAsFactors = FALSE, header = FALSE)
names(df) <- c('age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak',
str(df)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : num  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : num  1 1 1 1 0 1 0 0 1 1 ...
##  $ cp      : num  1 4 4 3 2 2 4 4 4 4 ...
##  $ trestbps: num  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : num  233 286 229 250 204 236 268 354 254 203 ...
##  $ fbs     : num  1 0 0 0 0 0 0 0 0 1 ...
##  $ restecg : num  2 2 2 0 2 0 2 0 2 2 ...
##  $ thalach : num  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : num  0 1 1 0 0 0 0 1 0 1 ...
```

```
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ slope   : num  3 2 2 3 1 1 3 1 2 3 ...
##  $ ca      : num  0 3 2 0 0 0 2 0 1 0 ...
##  $ thal    : num  6 3 7 3 3 3 3 3 7 7 ...
##  $ target  : int  0 2 1 0 0 0 3 0 2 1 ...
```

Prescindimos de la variable *ca* ya que no es una característica propia del cuerpo humano y no parece relevante para nuestro estudio.

```
df <- df[,-12]
```

Finalmente, en la fase previa a la limpieza de datos, tenemos un fichero con 303 observaciones y 13 variables.

3 Limpieza de los datos.

Empezamos esta sección observando una tabla-resumen con los estadísticos descriptivos de cada variable:

```
summarizeColumns(df) %>% knitr::kable(
  caption = "Resumen de estadíticos descriptivos")
```

Table 1: Resumen de estadíticos descriptivos

name	type	na	mean	disp	median	mad	min	max	nlevs
age	numeric	0	54.4389439	9.0386624	56.0	8.89560	29	77.0	0
sex	numeric	0	0.6798680	0.4672988	1.0	0.00000	0	1.0	0
cp	numeric	0	3.1584158	0.9601256	3.0	1.48260	1	4.0	0
trestbps	numeric	0	131.6897690	17.5997477	130.0	14.82600	94	200.0	0
chol	numeric	0	246.6930693	51.7769175	241.0	47.44320	126	564.0	0
fbs	numeric	0	0.1485149	0.3561979	0.0	0.00000	0	1.0	0
restecg	numeric	0	0.9900990	0.9949713	1.0	1.48260	0	2.0	0
thalach	numeric	0	149.6072607	22.8750033	153.0	22.23900	71	202.0	0
exang	numeric	0	0.3267327	0.4697945	0.0	0.00000	0	1.0	0
oldpeak	numeric	0	1.0396040	1.1610750	0.8	1.18608	0	6.2	0
slope	numeric	0	1.6006601	0.6162261	2.0	1.48260	1	3.0	0
thal	numeric	2	4.7342193	1.9397058	3.0	0.00000	3	7.0	0
target	integer	0	0.9372937	1.2285357	0.0	0.00000	0	4.0	0

En la tabla anterior observamos lo siguiente:
- Sólo una de las 13 variables contiene observaciones con valores perdidos (missing).
- La variable target tiene 4 valores, aunque sabemos que puede recodificarse en 2 categorías: el valor 0 indica que no hay enfermedad cardíaca, y el resto de valores sí indican la presencia de enfermedad cardíaca.
- En dicha tabla no se identifican variables categóricas, pero hay varias variables que claramente lo son, como por ejemplo sex o cp.
- A simple vista no se detectan valores atípicos en las variables que sabemos que son puramente numéricas, aunque más adelante lo comprobaremos con más detalle.

3.1 Tratamiento de valores perdidos

Tal y como vimos en la Table 1, la variable thal tiene 2 obseraciones con valores missing. Al tratarse sólo de 2 observaciones podríamos optar tanto por borrar dichos registros del dataset como imputarlos.

Por lo general, lo más adecuado sería proceder a la imputación de valores missing, ya que de esta manera no se pierde información. Sin embargo, hay que recordar que la imputación es un proceso que debe realizarse con cautela ya que puede afectar a los resultados obtenidos en los análisis, y realizarla de manera adecuada puede suponer una importante inversión de tiempo. En nuestro caso, al tratarse sólo de dos valores, los valores imputados a priori no van a alterar mucho los resultados. Para no perder esas dos observaciones, vamos a imputarlos por el método de kNN:

```
set.seed(1)
df$thal <- kNN(df)$thal
```

En caso de haber tenido más observaciones con valores missing, lo más adecuado sería considerar diferentes métodos de imputación alternativos (sustituir por medidas de tendencia central, kNN, modelos de regresión, imputación múltiple...), ver cuál es el más adecuado para nuestros datos y, a medida que se va avanzando en la imputación, realizar comprobaciones sobre los valores imputados para comprobar si afectan a aspectos fundamentales de los datos, como por ejemplo la distribución de las variables imputadas. Pero en nuestro caso no es necesario

Tras el tratamiento de valores missing, tenemos un dataset con 13 variables y 303 registros completos:

```
cat(c("Número de registros:", nrow(df), "\nNúmero de variables:", ncol(df)))
```

```
## Número de registros: 303
## Número de variables: 13
```

3.2 Variable *target*

Como hemos comentado anteriormente, la variable target presenta 4 valores que se pueden recodificar a valor 0 (no hay enfermedad cardíaca) o valor 1 (sí hay enfermedad cardíaca). Creamos la nueva variable *disease* a partir de *target*:

```
# Variable target2 para modelos
df$target2[df$target == 0] <- 0
df$target2[df$target == 1] <- 1
df$target2[df$target == 2] <- 1
df$target2[df$target == 3] <- 1
df$target2[df$target == 4] <- 1
# Variable target para exploración de datos
df$target <- factor(df$target2, labels = c("Sin enfermedad", "Con enfermedad"))
```

En la siguiente tabla tenemos la tabulación con las frecuencias absolutas de la variable *target*:

```
summary(df$target) %>% knitr::kable(caption = "Presencia de enfermedad del corazón",col.names = "Observa
```

Table 2: Presencia de enfermedad del corazón

	Observaciones
Sin enfermedad	164
Con enfermedad	139

3.3 Variables categóricas

Como hemos comentado anteriormente, en nuestro dataset hay algunas variables que se han cargado como numéricas, pero realmente se trata de variables categóricas. Estas variables son *sex*, *cp*, *fbs*, *restecg*, *exang*, *slope* y *thal*.

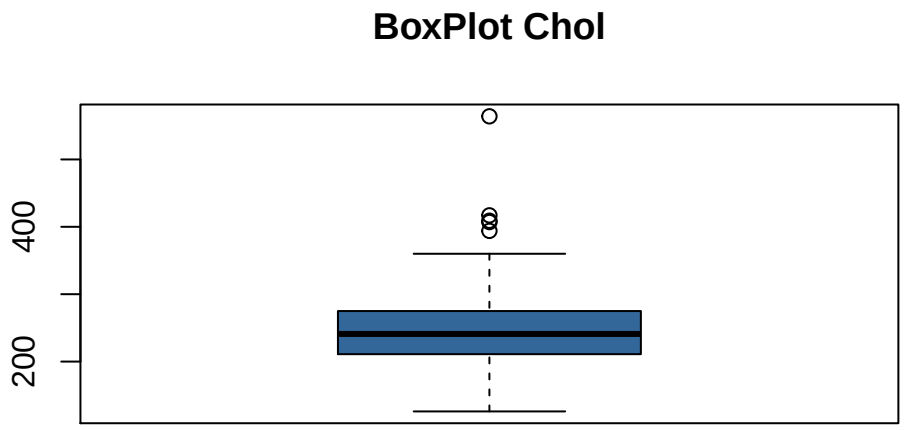
Siguiendo la información sobre categorías comentada en el apartado de descripción de las variables, transformamos estas variables en categóricas:

```
df <- df %>% mutate(sex = factor(sex, levels = c(0, 1), labels = c("Mujer", "Hombre")))
df <- df %>% mutate(cp = factor(cp, levels = c(1, 2, 3, 4), labels = c("tipico angina", "atipico angina
df <- df %>% mutate(fbs = factor(fbs, levels = c(0, 1), labels = c("Azucar en ayunas =< 120 mg/dl", "Az
df <- df %>% mutate(restecg = factor(restecg, levels = c(0, 1, 2), labels = c("Normal", "Anomalia en la
df <- df %>% mutate(exang = factor(exang, levels = c(0, 1), labels = c("No", "Sí")))
df <- df %>% mutate(slope = factor(slope, levels = c(1, 2, 3), labels = c("Creciente", "Plana", "Decrec
df <- df %>% mutate(thal = factor(thal, levels = c(3, 6, 7), labels = c("Normal", "Solucionado", "Rever
```

3.4 Identificación y tratamiento de valores extremos.

En los datos hay varias variables numéricas que presentan valores que pueden ser considerados como valores extremos, como por ejemplo la variable chol:

```
boxplot(df$chol, main="BoxPlot Chol", col="#336699")
```



Sin embargo, esto no quiere decir que haya que tratarlo. Eso dependerá del tipo de análisis que se quiera hacer, cómo de importante puede ser tener en cuenta valores extremos para dicho análisis y también del conocimiento previo sobre la materia. En el caso de este trabajo en particular puede ser relevante es relevante mantenerlos, ya que se busca detectar posibles patrones en los datos que puedan ayudar a diagnosticar el riesgo de padecer una enfermedad del corazón. Un valor extremo en alguna variable puede ser un indicador muy fuerte de que existe ese riesgo. Por esa razón no trataremos estos valores extremos.

```
write.csv(data, 'data/heart_disease_clean.csv')
```

4 Análisis de los datos.

El análosos de los datos se va a dividir en 3 ejercicios:

1) **Análisis exploratorio** de las variables sex, cp, thalach y thal, y su relación con la variable target
2) **Modelo de regresión logística** con target como variable dependiente
3) **Modelo de árbol de decisión** con target como variable dependiente

4.1 Análisis exploratorio de los datos y su relación con la presencia de enfermedad del corazón

4.1.1 sex

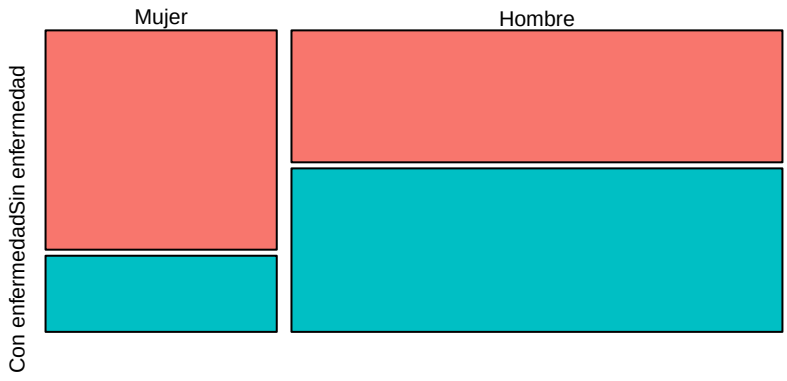
Calculamos la tabla de contingencia entre sex y target y la visualizamos gráficamente:

```
tc <- table(df$sex,df$target)
tc %>% kable(caption = "Tabla de contingencia de sex vs. target")
```

Table 3: Tabla de contingencia de sex vs. target

	Sin enfermedad	Con enfermedad
Mujer	72	25
Hombre	92	114

```
plot(tc, col = c("#F8766D","#00BFC4"), main = "sex vs. target")
```



En el grupo de hombres hay más pacientes con presencia de enfermedad del corazón. Vamos a comprobar si existe relación de independencia entre ambas variables. Para ello realizamos el test de independencia de chi-cuadrado entre sex y target:

```
chisq.test(tc, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tc
## X-squared = 23.218, df = 1, p-value = 1.446e-06
```

Existe dependencia entre ambas variables. Por tanto, parece que el sexo tiene una relación significativa con la presencia de enfermedad del corazón, y en concreto los hombres suelen presentar más presencia de enfermedad del corazón que las mujeres.

4.1.2 cp

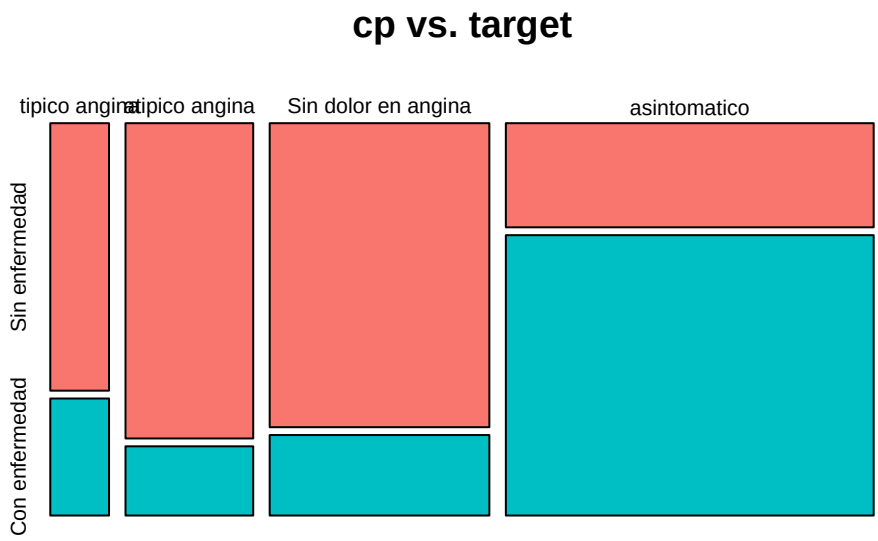
Calculamos la tabla de contingencia entre cp y target y la visualizamos gráficamente:

```
tc <- table(df$cp,df$target)
tc %>% kable(caption = "Tabla de contingencia de cp vs. target")
```

Table 4: Tabla de contingencia de cp vs. target

	Sin enfermedad	Con enfermedad
tipico angina	16	7
atipico angina	41	9
Sin dolor en angina	68	18
asintomatico	39	105

```
plot(tc, col = c("#F8766D","#00BFC4"), main = "cp vs. target")
```



L a mayoría de los pacientes no mostraron síntomas o dolor de angina. En el grupo de asintomáticos parece haber una proporción mucho mayor de pacientes que presentan enfermedad del corazón. En el resto de grupos no parece haber una diferencia muy grande. Esta observación la confirmamos haciendo un test de independencia de chi-cuadrado del grupo de asintomáticos frente al resto de grupos:

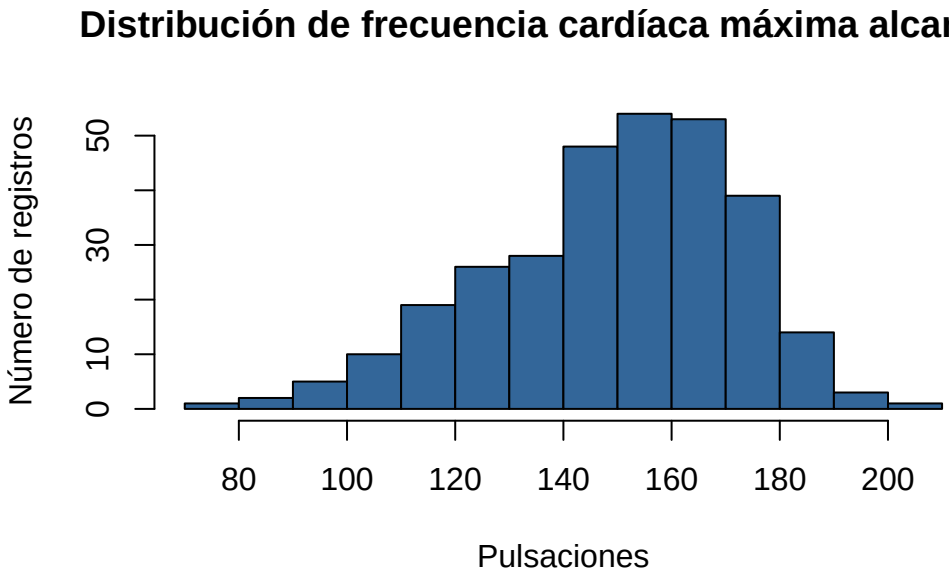
```
df$cp_asintomatico <- ifelse( df$cp == "asintomatico", "yes", "no")
tc_cpasint <- table(df$cp_asintomatico,df$target)
chisq.test(tc_cpasint, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tc_cpasint
## X-squared = 80.819, df = 1, p-value < 2.2e-16
```

4.1.3 thalach

Visualizamos el histograma de la frecuencia cardíaca:

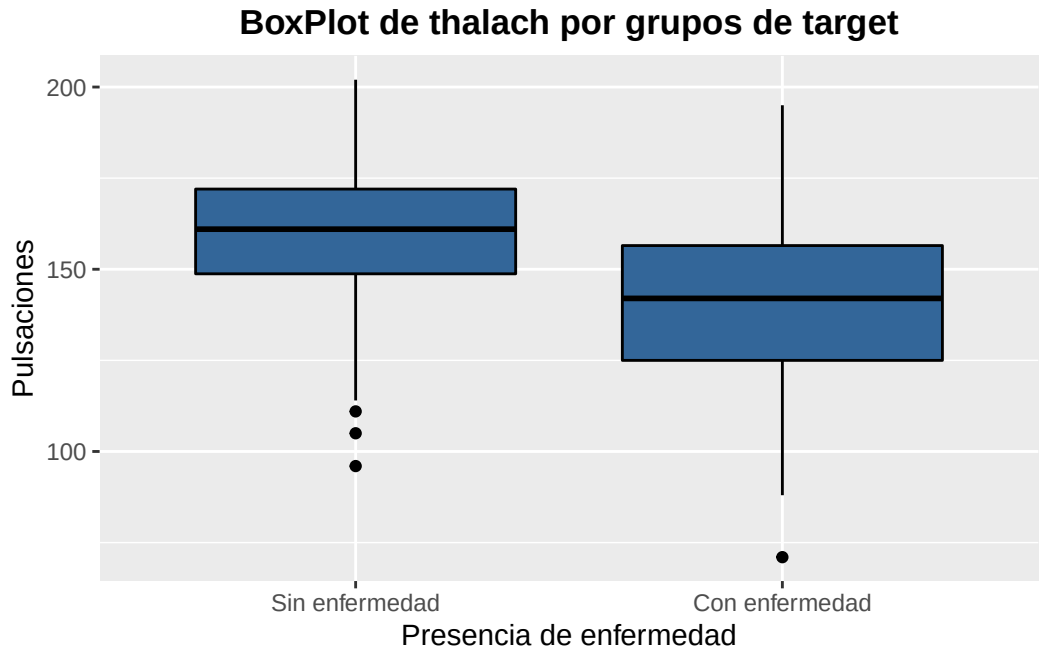
```
hist(df$thalach,col = "#336699",border = "black",prob = FALSE,xlab = "Pulsaciones",ylab = "Número de reg
```



En el histograma vemos que su distribución es ligeramente asimétrica y escorada hacia la parte superior se su distribución.

Ahora visualizamos los diagramas de caja de su distribución para cada grupo de target:

```
ggplot(df, aes(x = target, y = thalach))+geom_boxplot(color="#000000", fill="#336699")+ggtitle("BoxPlot
```



Los diagramas de caja sugieren que la frecuencia cardíaca de los pacientes que no presentan enfermedad es mayor que la de los pacientes que sí presentan enfermedad del corazón.

Testeamos la normalidad de las submuestras de target y la homogeneidad de sus varianzas:

```
# Test de normalidad para target = 1
shapiro.test(subset(df, target2==1)$thalach)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(df, target2 == 1)$thalach
## W = 0.98915, p-value = 0.3523
```

```
# Test de normalidad para target = 0
shapiro.test(subset(df, target2==0)$thalach)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(df, target2 == 0)$thalach
## W = 0.96659, p-value = 0.0005433
```

```
# Test de homogeneidad de varianzas
fligner.test(thalach ~ target, data = df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  thalach by target
## Fligner-Killeen:med chi-squared = 5.3987, df = 1, p-value = 0.02015
```

No ha normalidad en la distribución de frecuencia cardíaca en la submuestra de pacientes sin enfermedad y tampoco homogeneidad en las varianzas. Por tanto aplicamos el test de Wilcoxon para comprobar la igualdad de medias entre los grupos de pacientes con enfermedad y sin enfermedad:

```
wilcox.test(df$thalach ~ df$target, alternative = "greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df$thalach by df$target
## W = 16990, p-value = 9.305e-14
## alternative hypothesis: true location shift is greater than 0
```

El resultado del test concluye que la frecuencia cardíaca de los pacientes que no presentan enfermedad son significativamente mayores que la de los pacientes que sí presentan enfermedad cardíaca.

4.1.4 thal

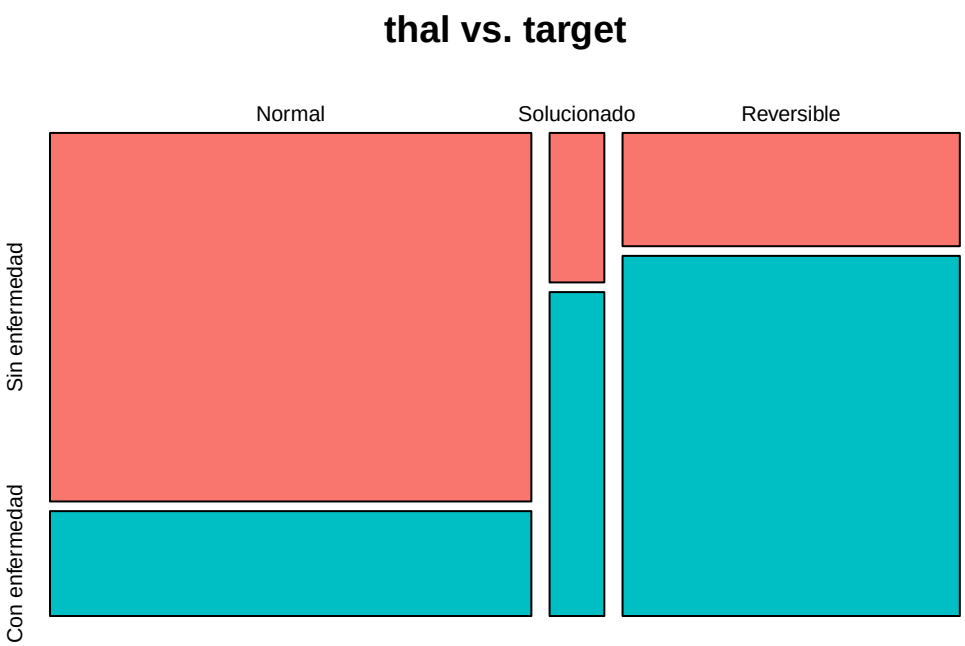
Calculamos la tabla de contingencia entre thal y target y la visualizamos gráficamente:

```
tc <- table(df$thal,df$target)
tc %>% kable(caption = "Tabla de contingencia de thal vs. target")
```

Table 5: Tabla de contingencia de thal vs. target

	Sin enfermedad	Con enfermedad
Normal	130	37
Solucionado	6	13
Reversible	28	89

```
plot(tc, col = c("#F8766D","#00BFC4"), main = "thal vs. target")
```



La mayoría de pacientes, en concreto 167, obtuvieron un resultado normal en la prueba de esfuerzo con Talio. También se ve que los grupos con resultado Solucionado y Reversible presentan más pacientes con enfermedad del corazón que los de resultado Normal. Visto esto, realizaremos un test de independencia chi-cuadrado entre thal y target para el grupo de resultado normal frente al resto de grupos:

```
df$thal_nonormal <- ifelse( df$thal != "Normal", "yes", "no")
tc_thalnonormal <- table(df$thal_nonormal,df$target)
chisq.test(tc_thalnonormal, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
```

```
##
## data:  tc_thalnonormal
## X-squared = 84.302, df = 1, p-value < 2.2e-16
```

El resultado del test confirma que existe dependencia entre target y thal, y en concreto que los pacientes que presentan un resultado Normal en la prueba de esfuerzo con Ralio suelen no tener enfermedad del corazón.

4.2 Regresión Logística

Planteamos un modelo de regresión logística con target2 (presencia de enfermedad del corazón) como variable dependiente del modelo y como variables independientes las otras 12 variables del dataset.

Separamos el conjunto de datos en muestras de entrenamiento y test. Dejaremos el 75% de la muestra para el conjunto de entrenamiento y el 25% restante para el conjunto de test.

```
set.seed(123)
split=sample.split(df$target2, SplitRatio = 0.75)
qualityTrain=subset(df,split == TRUE)
qualityTest=subset(df,split == FALSE)
nrow(qualityTrain)
```

```
## [1] 227
```

```
nrow(qualityTest)
```

```
## [1] 76
```

4.2.1 Estimación del modelo

Estimamos el modelo logit:

```
datasetlog=glm(target2 ~ age+sex+cp+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope+thal, data=qua
summary(datasetlog)
```

```
##
## Call:
## glm(formula = target2 ~ age + sex + cp + trestbps + chol + fbs +
##     restecg + thalach + exang + oldpeak + slope + thal, family = binomial,
##     data = qualityTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6896  -0.4802  -0.1352   0.4070   2.3705
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -6.837e+00  3.463e+00  -1.974 0.048364 *
## age                            3.766e-02  2.887e-02   1.305 0.192014
## sexHombre                      2.046e+00  6.160e-01   3.322 0.000894 ***
## cpatipico angina               1.065e+00  9.745e-01   1.093 0.274330
## cpSin dolor en angina          7.059e-01  8.998e-01   0.785 0.432731
```

```
## cpasintomatico                   2.704e+00  8.870e-01   3.049 0.002296 **
## trestbps                         2.418e-02  1.334e-02   1.813 0.069858 .
## chol                             4.625e-03  4.291e-03   1.078 0.281150
## fbsAzucar en ayunas > 120 mg/dl  2.124e-01  6.583e-01   0.323 0.746930
## restecgAnomalia en la onda ST-T  1.467e+01  1.336e+03   0.011 0.991237
## restecgHipertrofia Ventricular   4.371e-01  4.579e-01   0.955 0.339781
## thalach                         -2.973e-02  1.338e-02  -2.222 0.026297 *
## exangSí                          9.969e-02  5.216e-01   0.191 0.848425
## oldpeak                          5.668e-01  2.701e-01   2.099 0.035849 *
## slopePlana                       3.648e-01  5.467e-01   0.667 0.504674
## slopeDecreciente                -1.254e+00  1.350e+00  -0.929 0.353116
## thalSolucionado                  7.102e-01  9.444e-01   0.752 0.452064
## thalReversible                   1.498e+00  4.805e-01   3.117 0.001826 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 313.10  on 226  degrees of freedom
## Residual deviance: 148.37  on 209  degrees of freedom
## AIC: 184.37
##
## Number of Fisher Scoring iterations: 15
```

En este caso, como se ha podido comprobar existen muchas variables que no son significativas al 5%. Las eliminaremos de nuestro modelo. Las variables que eliminamos en las siguientes iteraciones son age, trestbps, chol, fbs, restecg y slope. Tras esto, estimamos de nuevo el modelo con la nueva especificación:

```
datasetlog_end=glm(target2 ~ sex+cp+thalach+oldpeak+thal, data=qualityTrain, family = binomial)
summary(datasetlog_end)
```

```
##
## Call:
## glm(formula = target2 ~ sex + cp + thalach + oldpeak + thal,
##     family = binomial, data = qualityTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5064  -0.4857  -0.1931   0.4660   2.6024
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.69567    1.80415   0.386  0.69980
## sexHombre              1.41936    0.49975   2.840  0.00451 **
## cpatipico angina       0.59785    0.89336   0.669  0.50335
## cpSin dolor en angina  0.43001    0.77791   0.553  0.58042
## cpasintomatico         2.32086    0.76091   3.050  0.00229 **
## thalach               -0.03056    0.01076  -2.842  0.00449 **
## oldpeak                0.65958    0.22542   2.926  0.00343 **
## thalSolucionado        0.96837    0.80420   1.204  0.22853
## thalReversible         1.73812    0.43370   4.008 6.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 313.10  on 226  degrees of freedom
## Residual deviance: 161.15  on 218  degrees of freedom
## AIC: 179.15
##
## Number of Fisher Scoring iterations: 5
```

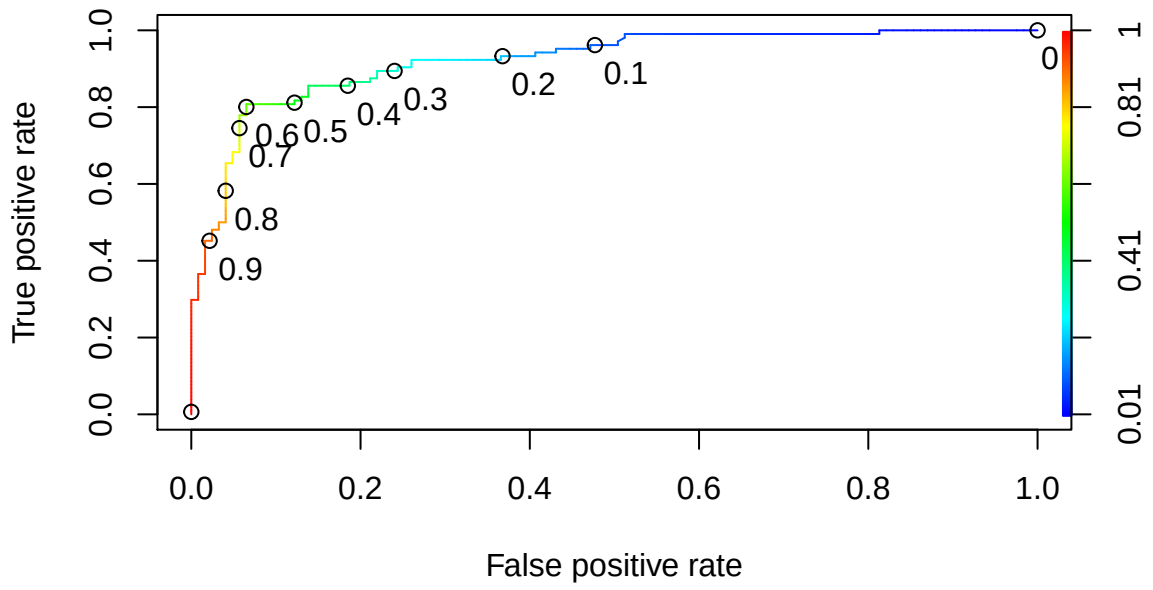
El AIC de este último modelo es 179.15, que es inferior al AIC del modelo original, que era 184.37. Esto hace que este segundo modelo de regresión logísta reducido sea preferible al extendido.

De este modelo logit se detectan 4 factores de riesgo (OR superiores a 1): el sexo del paciente (sex), el tipo de dolor en el pecho (cp), la depresión del ST inducida por el ejercicio en relación con el reposo (oldpeak), y el resultado de la prueba de esfuerzo con Talio (thal). Por otro lado, se detecta un factor de protección (OR inferior a 1) en la frecuencia cardíaca máxima alcanzada en la prueba de esfuerzo con Talio (thallach).

4.2.2 Curva ROC y Accuracy

Ahora analizamos como de bueno es nuestro modelo para entender la capacidad predictora que tiene. Para ello utilizamos un método muy común que es la representación de la Curva ROC:

```
# Predicciones del modelo
predictTrain=predict(datasetlog_end, type="response")
# Curva ROC
ROCRpred=prediction(predictTrain, qualityTrain$target2)
ROCRperf=performance(ROCRpred,'tpr','fpr')
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))
```



Teniendo en cuenta la Curva ROC obtenida, un umbral de clasificación de 0.6 parece ser bastante bueno, ya que parece que es el que corresponde al punto de la curva ROC que está más alejado de la diagonal.

Ahora medimos el Área Bajo la Curva (AUC):

```
auc = as.numeric(ROCR::performance(ROCRpred, 'auc')@y.values)
auc
```

```
## [1] 0.921318
```

El AUC de nuestro modelo es 0.92, lo cual indica que el modelo discrimina de manera excelente.

Finalmente, vamos a medir la accuracy de nuestro modelo tomando 0.6 como umbral de clasificación:

```
predictTest=predict(datasetlog, newdata = qualityTest,type = "response")
confMatrix <- table(qualityTest$target,predictTest >=0.6)
confMatrix %>% kable( caption = "Matrix de confusión modelo logit (0.6)")
```

Table 6: Matrix de confusión modelo logit (0.6)

	FALSE	TRUE
Sin enfermedad	35	6
Con enfermedad	12	23

```
accuracy <- (confMatrix[1]+confMatrix[4])/sum(confMatrix)
cat(c("Accuracy del modelo logit: ", accuracy))
```

```
## Accuracy del modelo logit:  0.763157894736842
```

En este caso el total de aciertos es de 58 de un total 76, lo que supone un accuracy del 76,32% con el dataset de test del que disponemos.

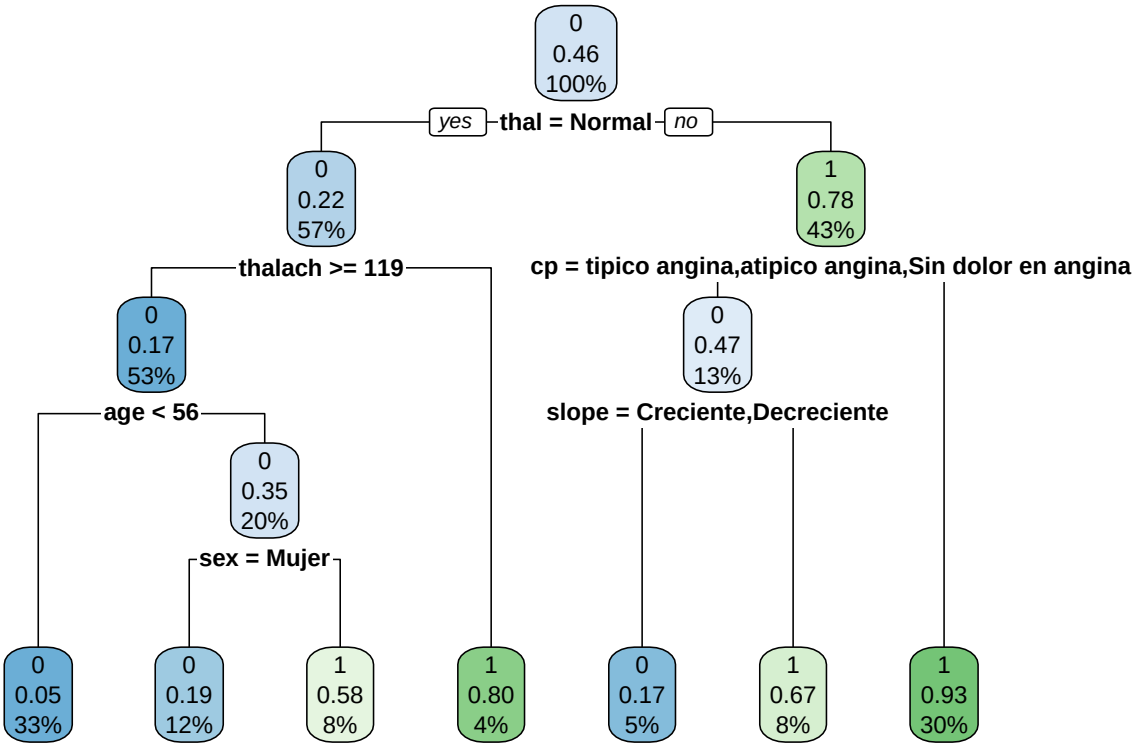
4.3 Árbol de decisión

Ahora planteamos un modelo basado en un árbol de decisión, de nuevo con target2 como variable dependiente.

4.3.1 Construcción del árbol de decisión

Creamos el modelo y dibujamos el árbol de decisión:

```
tree<-rpart(target2 ~ age+sex+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+ slope+cp+thal,method = "
rpart.plot(tree)
```



En este caso, podemos ver que las variables que el modelo tiene en cuenta a la hora de clasificar a los pacientes son thal, thalach, cp, age, slope y sex.

4.3.2 Análisis del árbol de decisión

Para analizar el modelo calculamos su matriz de confusión y obtenemos su accuracy:

```
# Predicciones con la muestra test
qualityTest$pred <- predict(tree, qualityTest,type="class")
qualityTest$target2 <- as.factor(qualityTest$target2)
# Matriz de confusión
confusionMatrix(qualityTest$pred, qualityTest$target2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 24  8
##          1 17 27
##
##                Accuracy : 0.6711
##                  95% CI : (0.5537, 0.7746)
##     No Information Rate : 0.5395
##     P-Value [Acc > NIR] : 0.01364
##
```

```
##                   Kappa : 0.3502
##
##  Mcnemar's Test P-Value : 0.10960
##
##             Sensitivity : 0.5854
##             Specificity : 0.7714
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.6136
##              Prevalence : 0.5395
##          Detection Rate : 0.3158
##    Detection Prevalence : 0.4211
##       Balanced Accuracy : 0.6784
##
##        'Positive' Class : 0
##
```

En esta ocasión estamos obteniendo un accuracy para nuestro dataset de test del 67.11%, algo más de 9 puntos porcentuales por debajo que el del modelo logit del apartado anterior.

4.3.3 Mejora del modelo: Random Forest

Como mejora al arbol de decisión vamos a plantear el uso de un modelo Random Forest que, en general, presenta mejores resultados que el obtenido por un árbol de decisión simple.

```
qualityTrain$target2 <- as.factor(qualityTrain$target2)
set.seed(100)
model_rf<-randomForest(target2 ~ age+sex+trestbps+chol+fbs+restecg+thalach+ exang+oldpeak+slope+cp+thal
model_rf
```

```
##
## Call:
##  randomForest(formula = target2 ~ age + sex + trestbps + chol +      fbs + restecg + thalach + exang
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 16.74%
## Confusion matrix:
##     0  1 class.error
## 0 106 17   0.1382114
## 1  21 83   0.2019231
```

En este caso el algoritmo de Random forest ha utilizado un total de 500 arboles utilizando un 3 variables en cada split.

4.3.4 Análisis del Random Forest

Al igual que con el modelo de árbol de decisión, obtenemos la matriz de confusión del Random Forest y calculamos su accuracy:

```r
# Predicciones con la muestra test
qualityTest$pred <- predict(model_rf, qualityTest,type="class")
# Matriz de confusión
confusionMatrix(qualityTest$pred, qualityTest$target2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 26  9
##          1 15 26
##
##                Accuracy : 0.6842
##                  95% CI : (0.5675, 0.7861)
##     No Information Rate : 0.5395
##     P-Value [Acc > NIR] : 0.007238
##
##                   Kappa : 0.3723
##
##  Mcnemar's Test P-Value : 0.307434
##
##             Sensitivity : 0.6341
##             Specificity : 0.7429
##          Pos Pred Value : 0.7429
##          Neg Pred Value : 0.6341
##              Prevalence : 0.5395
##          Detection Rate : 0.3421
##    Detection Prevalence : 0.4605
##       Balanced Accuracy : 0.6885
##
##        'Positive' Class : 0
##
```

Con el Random Forest hemos obtenido una mejora del accuracy con respecto al modelo del árbol de decisión simple, subiendo de 67.11% a 68.42%. Sin embargo, el accuracy de este modelo sigue siendo inferior al obtenido por el modelo de regresión logística, por lo que en caso de elegir alguno para hacer una clasificiación, nos quedaríamos con el modelo de regresión logística.

5 Conclusiones

El análisis de las variables sex, cp, thalach y thal nos ha llevado a la conclusión de que la presencia de enfermedad del corazón es más prevalente en hombres que en mujeres (sex), más en pacientes que no presentan síntomas de dolor en el pecho frente a otros que sí la presentan (cp), en pacientes con menor frecuencia cardíaca (thalach) y finalmente también entre los que presentan unos resultados de Solucionado o Reversible en la prueba de esfuezo con Talio (thal).

Del modelo de regresión logística se obtienen conclusiones similares. Se han detectado cuatro como factores de riesgo (sexHombre, cpasintomatico, oldpeak y thalReversible), y un factor de protección (tallach). La AUC del modelo es 0.92 y, se calculó una accuracy de 76.32% con un umbral de clasificación de 0.6.

Finalmente, en el modelo de árbol de decisión las variables que han sido relevantes para clasificar que un paciente tenia enfermedad del corazón han sido en primer lugar el valor del resultado de la prueba de esfuerzo con Talio (thal), la presión arterial (thalach), la edad del paciente (age), el sexo (sex), el tipo de dolor en el pecho, en concreto ser asintomático (cp), y la pendiente del segmento ST (slope). Adicionalmente se ha obtenido una mejora de este árbol de decisión utilizando el algoritmo de Random Forest, pero la accuracy de ambos modelos de clasifiación no superaba el 69%, por lo que el modelo de regresión logística es preferible frente a estos dos modelos.

6 Código y Dataset

El código fuente y el dataset utilizado para esta práctica pueden encontrarse en el repositorio Git accesible a través de este enlace: *Git*.

7 Tabla de contribuciones

Contribuciones	Firma
Investigación previa	R.L.G y C.G.C
Redacción de las respuestas	R.L.G y C.G.C
Desarrollo del código	R.L.G y C.G.C