Tipología y ciclo de vida de los datos - Práctica 2

Diego Castillo Carrión / Carlos Hernandez Martínez

Junio 2016

Carga de librerías necesarias para el desarrollo de la práctica.

Descripción del Data set

El dataset escogido para el desarrollo de la práctica se denomina *Heart Disease Data Set* o conjunto de datos de enfermedades cardiacas. Este dataset se encuentra disponible en repositorio de datos de Kaggle en el siguiente enlace.

El conjunto de datos consta de un total de 14 atributos entre discretos y contínuos que recojen información básica de pacientes como edad, sexo y también almacena el resultado de un conjunto de exámenes realizados, obteniendo por ejemplo el nivel de colesterol en la sangre, nivel de azúcar en la sangre, etc.

Los atributos que conforman el conjunto de datos son:

- *age* Edad del paciente
- *sex* Sexo del paciente
- *cp* Tipo de dolor en el pecho
- *trestbps* Presión arterial en reposo
- *chol* Suero colestoral
- *fbs* Glucemia en ayunas
- *restecg* Resultados electrocardiográficos en reposo
- *thalach* Ritmo cardiaco máximo alcanzado
- *exang* Angina inducida por el ejercicio
- *oldpeak* Depresión ST inducida por el ejercicio en relación con el descanso
- *slope* La pendiente del segmento pico del ejercicio ST
- *ca* Número de vasos principales (0-3) coloreados por fluoroscopia
- *thal* 3 = normal; 6 = defecto fijo; 7 = defecto reversible
- *target* Variable predictora

El conjunto de datos contiene información relevante de pacientes acerca de su actividad cardiaca recogidos mediante exámenes, así como también atributos como sexo y edad. Esta información permite resolver el problema de la detección temprana de enfermedades del corazón utilizando los atributos del dataset que permitan realizar la predicción de estas enfermedades.

Integración y selección de los datos de interés a analizar.

Se realiza la carga del conjunto de datos y se consulta una porción de los mismos con el fin de observar si estos se han cargado de manera correcta.

```
# Se fija el espacio de trabajo
setwd("C:/Users/dell/Dropbox/maestria/Tipología y ciclo de vida de los datos/Práctica 2")

# Carga de los datos
heart <- read.csv('heart.csv', header = FALSE)
names(heart) <- c("age","sex","cp","trestbps","chol","fbs","restecg","thalach","exang","oldpeak","slope
```

```r
# Revisión de los datos
head(heart)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3  41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

Se utiliza el comando `summary` con el fin de observar algunas aestadísticas básicas del dataset.

```r
# Realizamos un análisis preliminar de los datos
summary(heart)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0
##       chol            fbs            restecg          thalach
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :240.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.3   Mean   :0.1485   Mean   :0.5281   Mean   :149.6
##  3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##      exang           oldpeak         slope             ca
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000
##       thal           target
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:2.000   1st Qu.:0.0000
##  Median :2.000   Median :1.0000
##  Mean   :2.314   Mean   :0.5446
##  3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000
```

Se puede observar que el dataset contiene información de personas de 29 a 77 años con una media de edad de 55 años, la varaible *thalach* presenta el ritmo cardiaco máximo alcanzado por los pacientes cuyos valores van desde 71 a 202 pulsasiones por minuto, etc.

Uitlizamos también el comando `str` para observar la estructura del data set.

```
# Revisamos de nuevo la estructura del dataset
str(heart)
```

```
## '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 ...
```

Como resultado se tiene que todas las variables son numéricas, aunque algunas de ellas son discretas y se debe convertir en factores, como la variable `sex` que contiene valores que corresponden a 0= mujer y 1 = hombre, la variable `cp` cuyos valores van de 0 a 3 dependiendo del tipo del dolor en el pecho, el atributo `fbs` que toma el valor 0 si no presenta glucemia en ayunas y el valor 1 cuando si presenta, la variable `restecg` recoge los resultados de los exámenes electrocardigráficos en reposos que van de 0 a 3, la variable `exang` toma valores de 0 a 1 si se presenta angina inducidad por ejercicio, la variable `slope` toma valres de 0 a 2 de acuerdo a la pendiente del segmento del ejercicio, `ca` almacena el número de vasos principales que van desde 0 a 3, y `thal` cuyos valores van de 0 a 3 si presenta algún defecto.

Por lo visto, se realiza el cambio a factores de las variables discretas.

```
# Cambiamos los valores correspondiente a mujeres y hombres en la varaible sex
heart <- heart %>% mutate(sex=ifelse(sex==1,"hombre","mujer"))

# Factorizamos las variables discretas.
cols<-c("sex","cp","fbs","restecg","exang","slope","ca","thal","target")
for (i in cols){
  heart[,i] <- as.factor(heart[,i])
}
```

Luego de transformar los atributos a factores, revisamos de nuevo la estructura de los datos para confirmarlo.

```
str(heart)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : Factor w/ 2 levels "hombre","mujer": 1 1 2 1 2 1 2 1 1 1 ...
##  $ cp      : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ 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     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
##  $ restecg : Factor w/ 3 levels "0","1","2": 1 2 1 2 2 2 1 2 2 2 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
##  $ ca      : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ thal    : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
##  $ target  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

Limpieza de los datos.

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Al tratarse de valores numéricos, no se realizará el análisis o identificación de valores iguales a 0 ya que al revisar el significado de cada variable, nos damos cuenta que las variables que contienen valores iguales a 0 son resultados de exámenes aplicados a los pacientes, y que, según el caso pueden ser iguales a 0.

Los siguiente comandos realizan la indentificación de los valores que son nulos y también los valores que se encentran vacíos dentro del dataset.

```
# Identificamos valores nulos
sapply(heart,function(x) sum(is.na(x)))
```

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

```
# Se realiza la búsqueda de elementos que contengan valores vacios
colSums(heart =="")
```

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

Como resultado de los comandos ejecutados, se tiene que los atributos del dataset no contienen valores nulos ni tampoco valores vacios

En el supuesto caso en que se tuviera atributos que contengan valores nulos o vacíos, la eficacia de las técnicas de tratamiento de estos valores está directamente relacionada con la razón por la cual tuvo su origen el valor perdido. Si tenemos alguna información acerca de ella, es posible que encontremos una regla para completar estos valores, por el contrario, si no tenemos dicha información, es necesario aplicar técnicas de evaluación de los valores perdidos que encuentren algún patrón que permita ya sea completarlos o descartarlos(en el caso que no afecten el análisis),decisión que depende en gran medida del tipo del valor perdido y la importancia del registro en la base de datos (Allison,2001).

Identificación y tratamiento de valores extremos.

Se entiende por valores extremos o `outliers` a aquellas observaciones que se desvían mucho de otras observaciones y despierta sospechas de ser generadas por mecanismos diferentes.

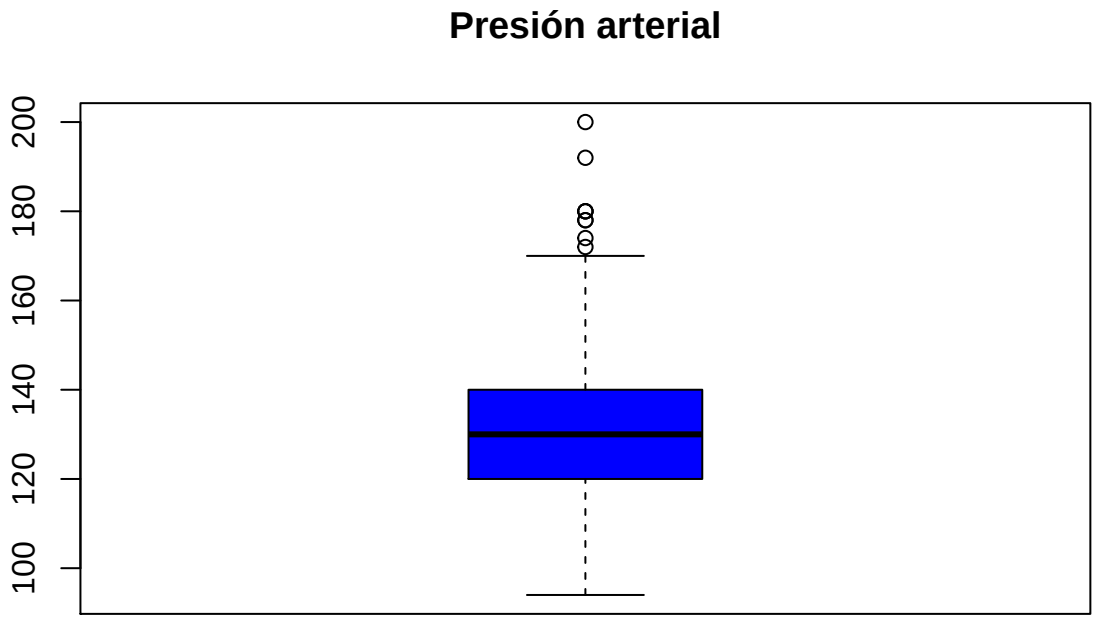
En la presente práctica analizaremos los siguientes atributos continuos en busca de outliers: `trestbps`, `chol`, `thalach` y `oldpeak`.

Nota: Al tratarse de resultados de exámenes realizados a pacientes, es muy normal encontrar valores desorbitados o fuera de lo normal en algunos de ellos, debido a diferentes causas como alimentación, ejercicio físico, estilo de vida, etc. Por ello, los outliers de este dataset no deberían ser corregidos. Únicamente por motivos didácticos correspondientes a la práctica se realizará la corrección de estos valores extremos

trestbps

Como se indicó anteriormente, el atributo `trestbps` hace referencia a la presión aterial en reposo del paciente, primero se grafica el conjunto de datos para este atributos, con el fin de detectar outliers.

```
# Graficamos en busca de outliers
boxplot(heart$trestbps,main = "Presión arterial",boxwex = 0.5,col="blue")
```



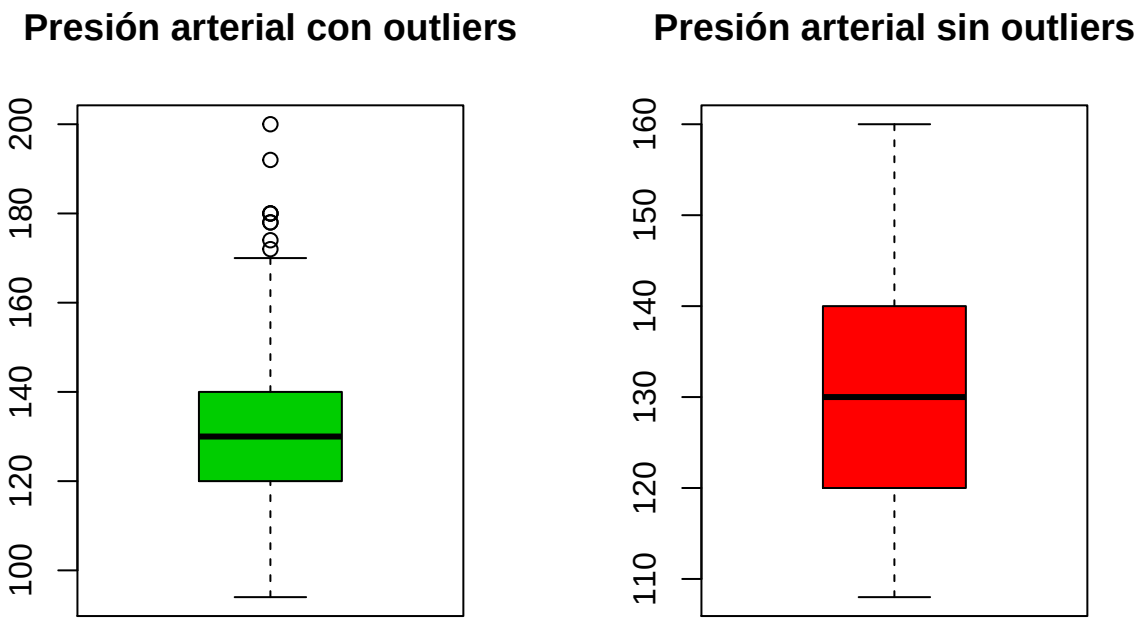
Se puede observar que existen valores extremos, por lo que se procede a su corrección, para estos casos vamos a imputar los outliers reemplazando el valor de los outliers por la media de la variable `trestbps`.

```
# Realizamos la imputación de los outliers
imputar_outliers <- function(x, removeNA = TRUE){
  quantiles <- quantile(x, c(0.05, 0.95), na.rm = removeNA)
  x[x<quantiles[1]] <- mean(x, na.rm = removeNA)
  x[x>quantiles[2]] <- median(x, na.rm = removeNA)
  x
}

trestbps_imputada <- imputar_outliers(heart$trestbps)
```

Por últimos, realizamos las comparación de los valores de la varaible `trestbps` con y sin outliers.

```
#Graficamos las diferencias
par(mfrow = c(1,2))
boxplot(heart$trestbps, main = "Presión arterial con outliers",
        col = 3)
boxplot(trestbps_imputada, main = "Presión arterial sin outliers",col=2)
```

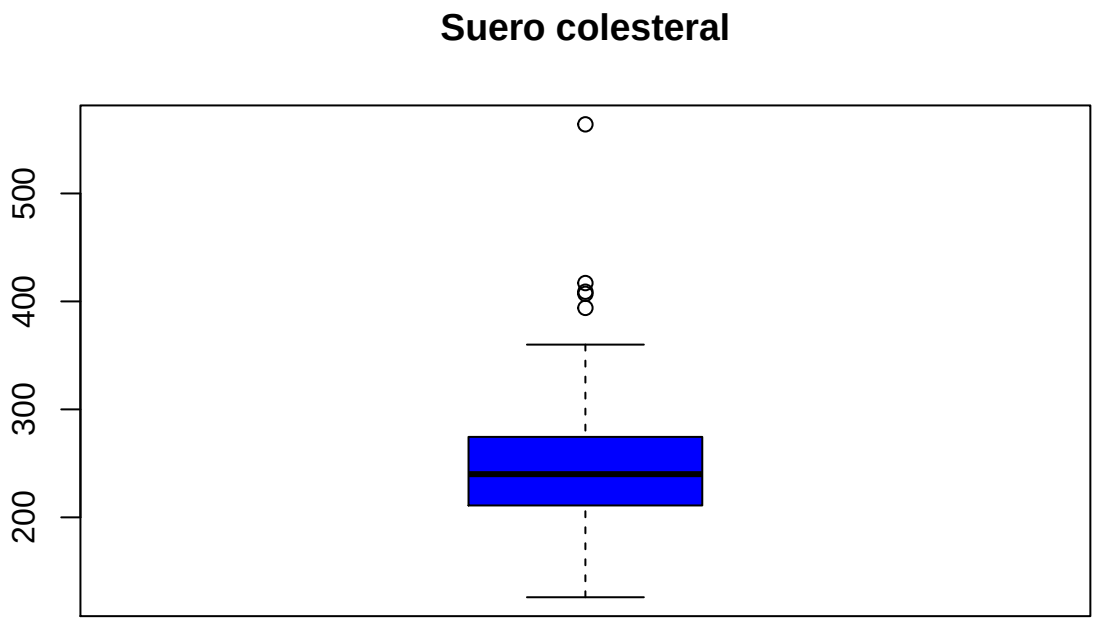



```
# Actualizamos los valores correctos
heart$trestbps <- trestbps_imputada
```

chol

Siguiendo con el desarrollo de la práctica, realizamos el análisis de la variable `chol`, la cual contiene el valor del suero colesteral de los pacientes. Primero revisamos si existen outliers.

```
# Graficamos en busca de outliers
boxplot(heart$chol,main = "Suero colesteral",boxwex = 0.5,col="blue")
```

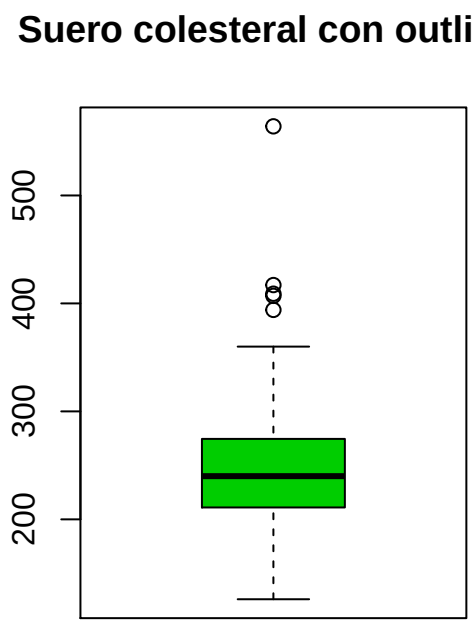


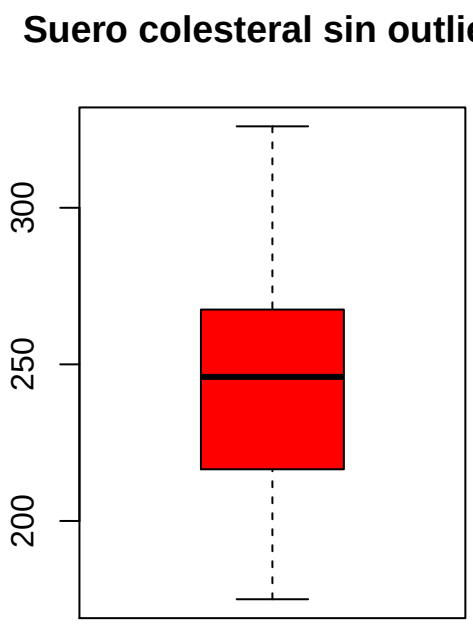
Al evidenciar que efectivamente se tienen valores extremos, se procede con la corrección de los mismos utilizando la función desarrollada en la anterior sección.

```
# Realizamos la imputación de los outliers
chol_imputada <- imputar_outliers(heart$chol)
```

Por últimos, realizamos las comparación de los valores de la varaible `chol` con y sin outliers.

```
#Graficamos las diferencias
par(mfrow = c(1,2))
boxplot(heart$chol, main = "Suero colesteral con outliers",
        col = 3)
boxplot(chol_imputada, main = "Suero colesteral sin outliers",col=2)
```



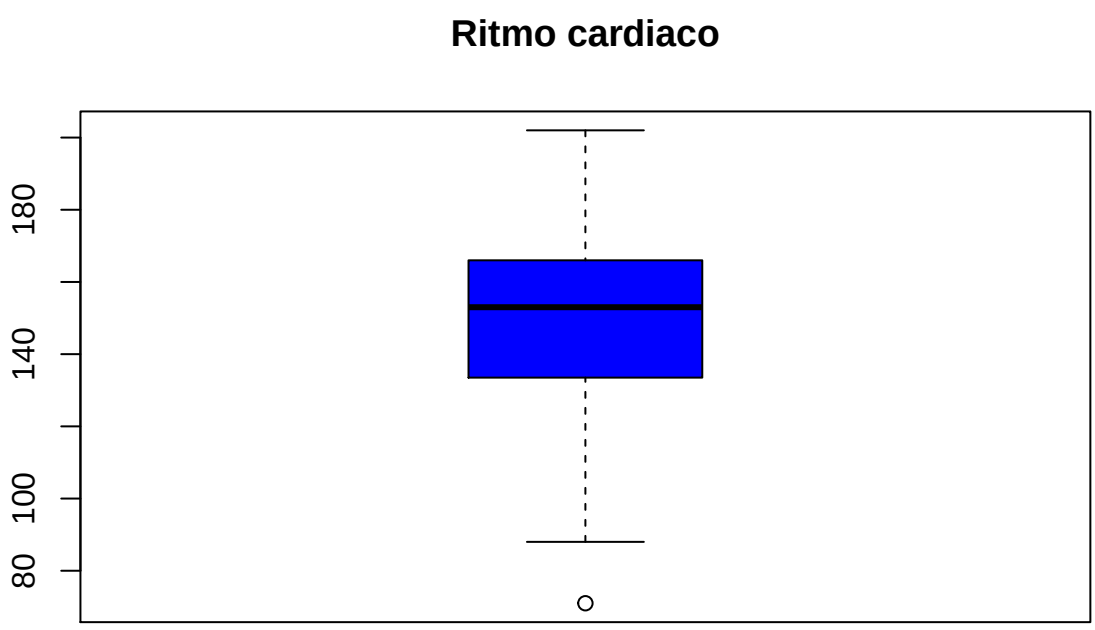



```
# Actualizamos los valores correctos
heart$chol <- chol_imputada
```

thalach

La variable thalach almacena el ritmo cardiaco máximo al canzado por los pacientes. en esta variable continua también se analiza si se cuenta con outliers.

```
# Graficamos en busca de outliers
boxplot(heart$thalach,main = "Ritmo cardiaco",boxwex = 0.5,col="blue")
```

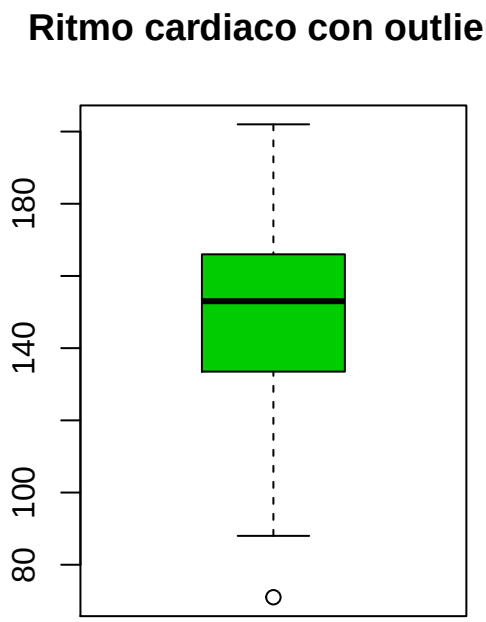


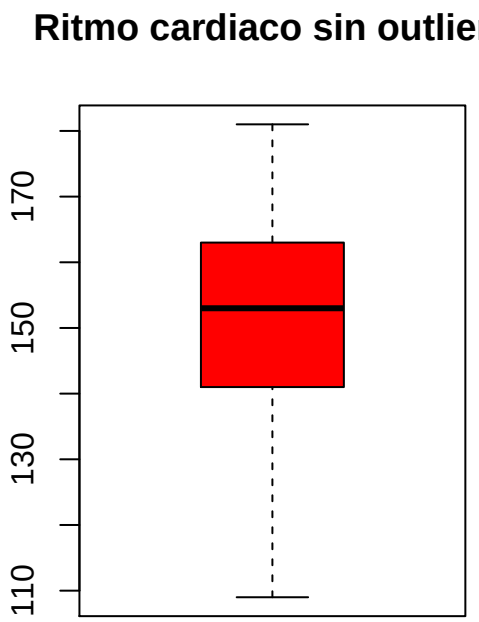
Solo existe un valor extremos, al cual también se corregirá.

```
# Realizamos la imputación de los outliers
thalach_imputada <- imputar_outliers(heart$thalach)
```

Se realiza comprobación de los datos con y sin outliers.

```
#Graficamos las diferencias
par(mfrow = c(1,2))
boxplot(heart$thalach, main = "Ritmo cardiaco con outliers",
        col = 3)
boxplot(thalach_imputada, main = "Ritmo cardiaco sin outliers",col=2)
```



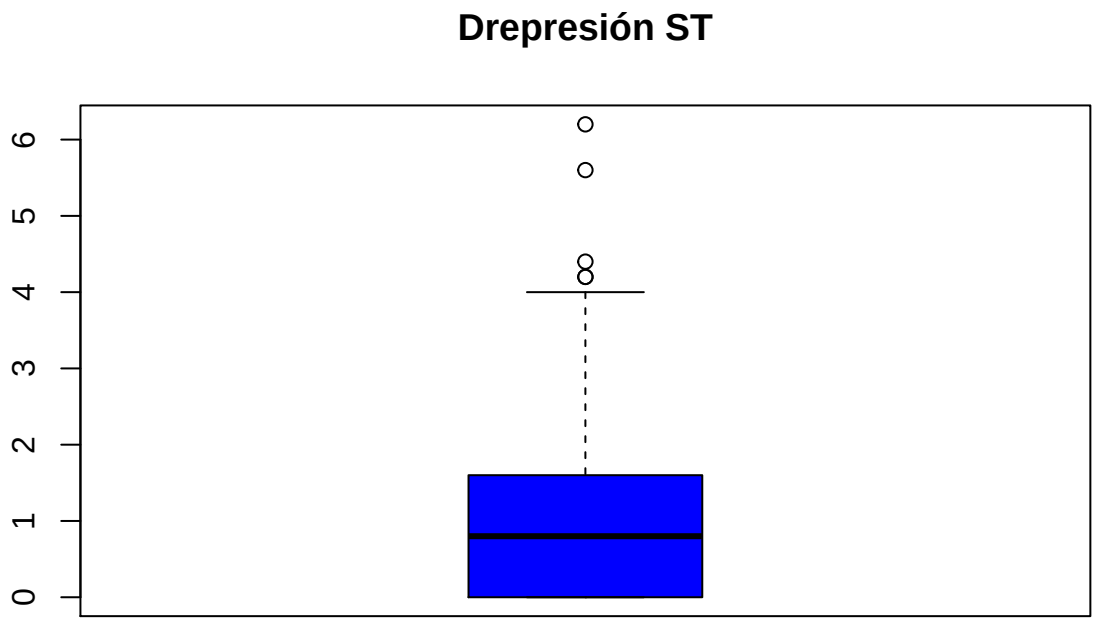



```
# Actualizamos los valores del atributo
heart$thalach <- thalach_imputada
```

oldpeak

Esta última variable continua contiene la depresión ST inducida por el ejercicio en relación con el descanso, y se procede con el análisis de la misma en busca de outiers.

```
# Graficamos en busca de outliers
boxplot(heart$oldpeak,main = "Drepresión ST",boxwex = 0.5,col="blue")
```

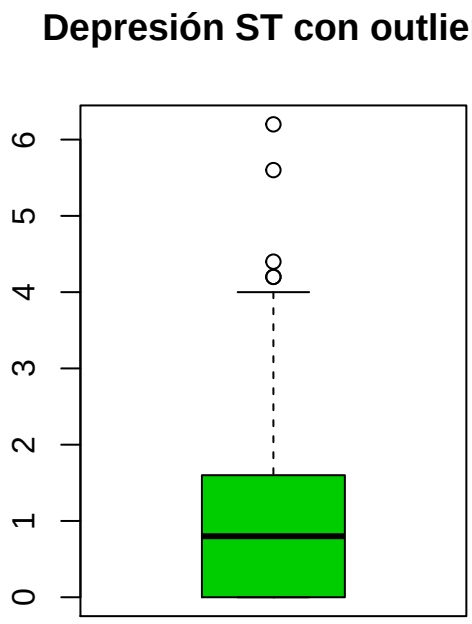


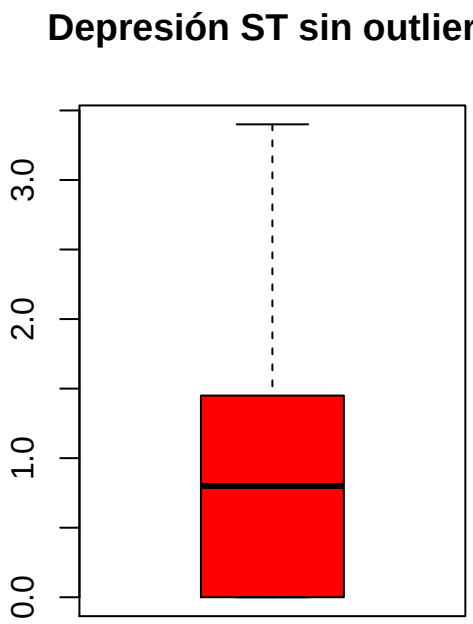
Se realiza la corrección de los outliers detectados, utilizando la imputación.

```
# Realizamos la imputación de los outliers
oldpeak_imputada <- imputar_outliers(heart$oldpeak)
```

Se realiza comprobación de los datos con y sin outliers.

```
#Graficamos las diferencias
par(mfrow = c(1,2))
boxplot(heart$oldpeak, main = "Depresión ST con outliers",
        col = 3)
boxplot(oldpeak_imputada, main = "Depresión ST sin outliers",col=2)
```






```
# Actualizamos los valores del atributo
heart$oldpeak <- oldpeak_imputada
```

Análisis de los datos

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

El dataset escogido tiene como fin utilizar un conjunto de tratributos de varios pacientes con el fin de predecir el desarrollo de enfermedades cardiacas. Por lo que los modelos construidos se basarán en la implementación de algoritmo de clasificación y regresión que nos permitan predecir estas enfermedades. para su desarrollo, los datos se deben dividir en dos partes una parte del 70% para el entrenamiento y el restante 30% para las validaciones.

La siguiente porción de código realiza la extracción de estos subconjuntos de datos.

```
set.seed(10)
inTrainRows <- createDataPartition(heart$target,p=0.7,list=FALSE)
trainData <- heart[inTrainRows,]
testData <-  heart[-inTrainRows,]
```

Almacenamos los datos resultantes de la limpieza en un nuevo archivo.

```
write.csv(heart,file = "heartClean.csv")
```

Comprobación de la normalidad y homogeneidad de la varianza

Comprobación de la normalidad

En muchos trabajos y publicaciones se ha visto que en los análisis estadísticos que se realizan, suponen que las variables continuas siguen una distribución normal sin antes realizar una verificación previa.

En la presente sección de sealizará la comprovación primero visual y luego mediante la aplicación del test de nomralidad denominado de `asimetría`, ya que es uno de los más recomendados.

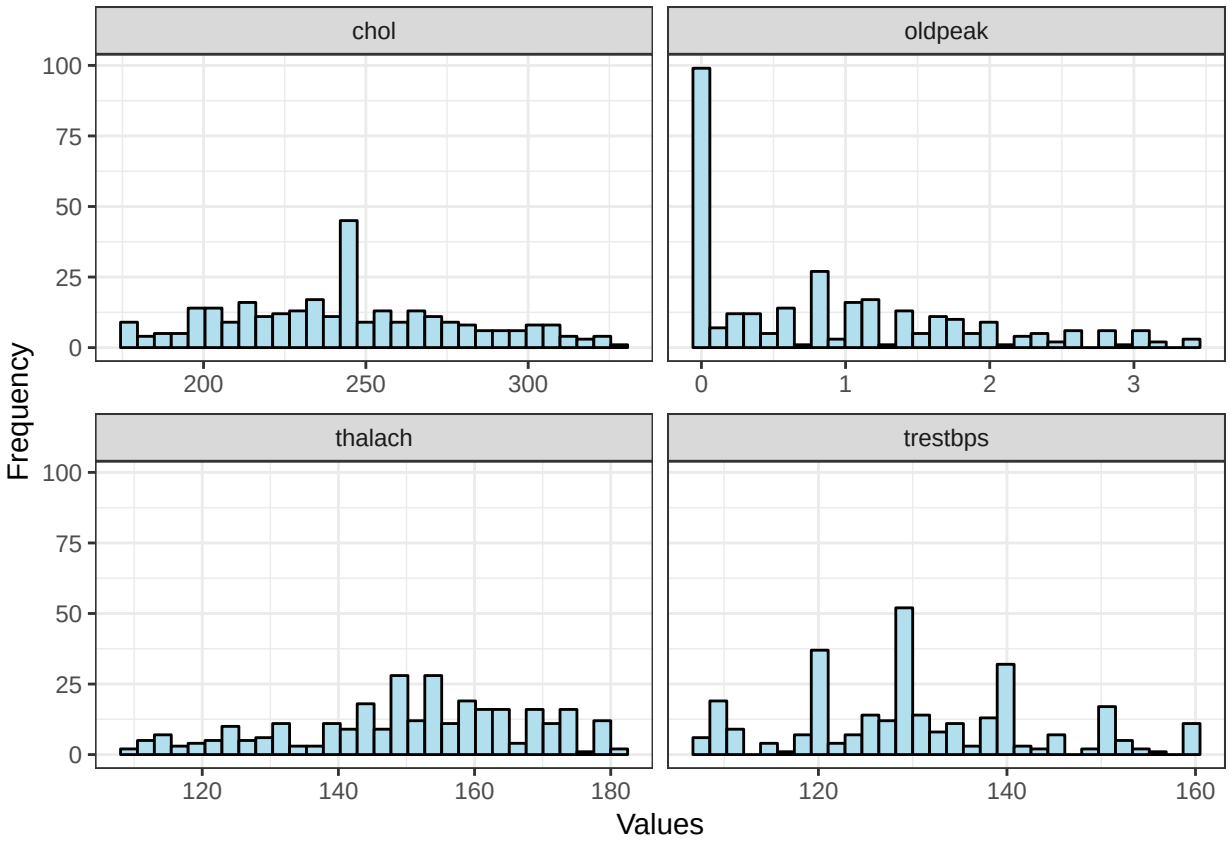
Las variables a las que se aplicarán estos test son:`trestbps`, `chol`, `thalach` y `oldpeak`.

Análisis visual

Utilizando un histograma se puede analizar de manera visual la distribución que siguen los datos y así determinar su normalidad.

```
heart %>%
  gather(Attributes, value, trestbps,chol,thalach,oldpeak) %>%
  ggplot(aes(x=value)) +
  geom_histogram(fill="lightblue2", colour="black") +
  facet_wrap(~Attributes, scales="free_x") +
  labs(x="Values", y="Frequency") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Se puede determinar que por lo menos visualmente, los atributos `chol` y `trstbps` siguen una distribución parecida a la normal.

Realizamos el test de normalidad sobre los atributos, utilizando el test de asimetría.

```
skewness(heart$trestbps)
```

```
## [1] 0.2841577
```

```
skewness(heart$chol)
```

```
## [1] 0.232408
```

```
skewness(heart$thalach)
```

```
## [1] -0.4555761
```

```
skewness(heart$oldpeak)
```

```
## [1] 0.8606549
```

Se logra determinar que ninguna de las variable sigue una distribución normal.

Homogeneidad de la varianza

El supuesto de homogeneidad de varianzas, conocido también como *homocedasticidad*, considera que la varianza es constante en los diferentes niveles de un factor, es decir, entre diferentes grupos.

Existen diferentes tests que permiten evaluar la distribución del a varianza. Todos ellos consideran como hipótesos nula que la varianza es igual entre los grupos y como hipotesis alternativa que no lo es.La diferencia entre ellos es el estádistico de centralidad que utilizan: media, mediana, media truncada. Een nuestro ejemplo las variables no tienen una distribución normal por lo que se utilizará el *Test de Levene*, cual se caracteriza por utilizar la mediana y por no depender de la distribución de las variables.

Se revisa la homogeneidad de los atribtos de tipo factores para los valores de la variable `trestbps`.

Para anlaizar los resultados obtenidos, realizamos la formulación de las hipótesis, en donde, la *Hipótesis nula* no dice que existe homogeneidad entre las varianzas, en cambio la *hipótesis alterna* indica que las varianzas son diferentes.

Para tomar la decisión nos basamos en el valor de `p`, en donde, si `p<0.05` entonces rechazamos la hipótesis nula y nos quedamos con la hipótesis del investigador.

```
# age
leveneTest(y = heart$trestbps, group = heart$age, center = "median")
```

```
## Warning in leveneTest.default(y = heart$trestbps, group = heart$age, center
## = "median"): heart$age coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group  40  0.9568 0.5492
##       262
```

```r
# Sex
leveneTest(y = heart$trestbps, group = heart$sex, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   1   0.403  0.526
##       301
```

```r
# cp
leveneTest(y = heart$trestbps, group = heart$cp, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   3  1.6699 0.1735
##       299
```

```r
# fbs
leveneTest(y = heart$trestbps, group = heart$fbs, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value  Pr(>F)
## group   1  4.7414 0.03022 *
##       301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# restecg
leveneTest(y = heart$trestbps, group = heart$restecg, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   2  0.4034 0.6684
##       300
```

```r
# exang
leveneTest(y = heart$trestbps, group = heart$exang, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   1  1.3848 0.2402
##       301
```

```r
# ca
leveneTest(y = heart$trestbps, group = heart$ca, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   4  1.6096 0.1718
##       298
```

```r
# slope
leveneTest(y = heart$trestbps, group = heart$slope, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   2  0.1408 0.8687
##       300
```

```r
# thal
leveneTest(y = heart$trestbps, group = heart$thal, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   3  1.4537 0.2273
##       299
```

Según los resultados obtenidos, se tiene que únicamente la variable `fbs` no cumple con el novel de significancia por lo que se considera que su varianza es diferente. En el resto de variables nos quedamos con la hipótesis nula o que existe homogeneidad de varianza.

Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo de estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos 3 métodos de análisis diferentes.

Haciendo una revisión del conjunto de datos, se observa que el objetivo del mismo es utilizar un conjunto de variables demográficas de pacientes como la edad y el sexo en conjunto de otros variutos resultantes de exámenes realizados para predecir si un paciente sufre de una enfermedad cardiaca.

Es por ello, que las pruebas estadísticas correspondientes al desarrollo de la práctica se enfocan en la creación de modelos de clasificación que permitan determinar si los atributos del dataset pueden predecir de una manera correcta las enfermedades cardiacas de los pacientes.

primero se implementará un modelo de regresión logística seguido por un modelo de clasificación utilizando el método random forest.

Regresión logística

La regresión logística es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de variables independientes o predictoras. En nuestro casos, la variable que se desea predecir es `target` que puede tomar los valores 0 cuando el paciente no presenta ninguna enfermedad o 1 cuando el el paciente presenta una enfermedad cardiaca.

```r
# Modelo de regresión logística
set.seed(10)
logRegModel <- train(target ~ ., data=trainData, method = 'glmnet', family = 'binomial')
logRegPrediction <- predict(logRegModel, testData)
logRegPredictionprob <- predict(logRegModel, testData, type='prob')[2]
logRegConfMat <- confusionMatrix(logRegPrediction, testData[,"target"])

logRegConfMat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 31  3
##          1 10 46
##
##                Accuracy : 0.8556
##                  95% CI : (0.7657, 0.9208)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 3.463e-10
##
##                   Kappa : 0.7047
##
##  Mcnemar's Test P-Value : 0.09609
##
##             Sensitivity : 0.7561
##             Specificity : 0.9388
##          Pos Pred Value : 0.9118
##          Neg Pred Value : 0.8214
##              Prevalence : 0.4556
##          Detection Rate : 0.3444
##    Detection Prevalence : 0.3778
##       Balanced Accuracy : 0.8474
##
##        'Positive' Class : 0
##
```

Utilizando el modelo de regresión logística se puede determinar que el conjunto de datos puede predecir en un **83,3%** la presencia de enfermedades cardiacas en los pacientes.

Clasificación utilizando el método random forest

Random Forest es un método de clasificación supervisada el cual es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

Para la implementación de este algoritmo de utilizarán los conjunto de prueba y test creados para la implementación de la regresión liogística anterior.

```
# Random Forest
set.seed(10)
RFModel <- randomForest(target ~ .,
                    data=trainData,
                    importance=TRUE,
                    ntree=2000)
RFPrediction <- predict(RFModel, testData)
RFPredictionprob = predict(RFModel,testData,type="prob")[, 2]
RFConfMat <- confusionMatrix(RFPrediction, testData[,"target"])
RFConfMat
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  0  1
##          0 32  5
##          1  9 44
##
##                Accuracy : 0.8444
##                  95% CI : (0.7528, 0.9123)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 1.629e-09
##
##                   Kappa : 0.6839
##
##  Mcnemar's Test P-Value : 0.4227
##
##             Sensitivity : 0.7805
##             Specificity : 0.8980
##          Pos Pred Value : 0.8649
##          Neg Pred Value : 0.8302
##              Prevalence : 0.4556
##          Detection Rate : 0.3556
##    Detection Prevalence : 0.4111
##       Balanced Accuracy : 0.8392
##
##        'Positive' Class : 0
##
```

de manera similar que en la aplicación de la regresión logística, utilizando el algoritmo de Random Forest se puede predecir en un **80%** las posibles enfermedades del corazón sobre los pacientes según los atributos del dataset.

Clasificación utilizando el método de árboles de decisión

```
# aplicamos el algoritmo de árboles de decisión
classificationTree3 <- rpart(target ~ ., data = trainData, method = "class")
print(classificationTree3)
```

```
## n= 213
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 213 97 1 (0.45539906 0.54460094)
##    2) thal=0,1,3 98 25 0 (0.74489796 0.25510204)
##      4) ca=1,2,3,4 58  5 0 (0.91379310 0.08620690) *
##      5) ca=0 40 20 0 (0.50000000 0.50000000)
##       10) exang=1 16  3 0 (0.81250000 0.18750000) *
##       11) exang=0 24  7 1 (0.29166667 0.70833333)
##         22) age< 50 9  4 0 (0.55555556 0.44444444) *
##         23) age>=50 15  2 1 (0.13333333 0.86666667) *
##    3) thal=2 115 24 1 (0.20869565 0.79130435)
##      6) cp=0,3 47 20 1 (0.42553191 0.57446809)
##       12) ca=1,2,3 19  5 0 (0.73684211 0.26315789) *
```

```
##       13) ca=0 28  6 1 (0.21428571 0.78571429) *
##      7) cp=1,2 68  4 1 (0.05882353 0.94117647) *
```

```
pred <- predict(classificationTree3,testData,type="class")
t <- table(testData$target,pred)
confusionMatrix(t)
```

```
## Confusion Matrix and Statistics
##
##    pred
##      0  1
##   0 33  8
##   1  7 42
##
##                Accuracy : 0.8333
##                  95% CI : (0.74, 0.9036)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : 2.25e-08
##
##                   Kappa : 0.6633
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8250
##             Specificity : 0.8400
##          Pos Pred Value : 0.8049
##          Neg Pred Value : 0.8571
##              Prevalence : 0.4444
##          Detection Rate : 0.3667
##    Detection Prevalence : 0.4556
##       Balanced Accuracy : 0.8325
##
##        'Positive' Class : 0
##
```

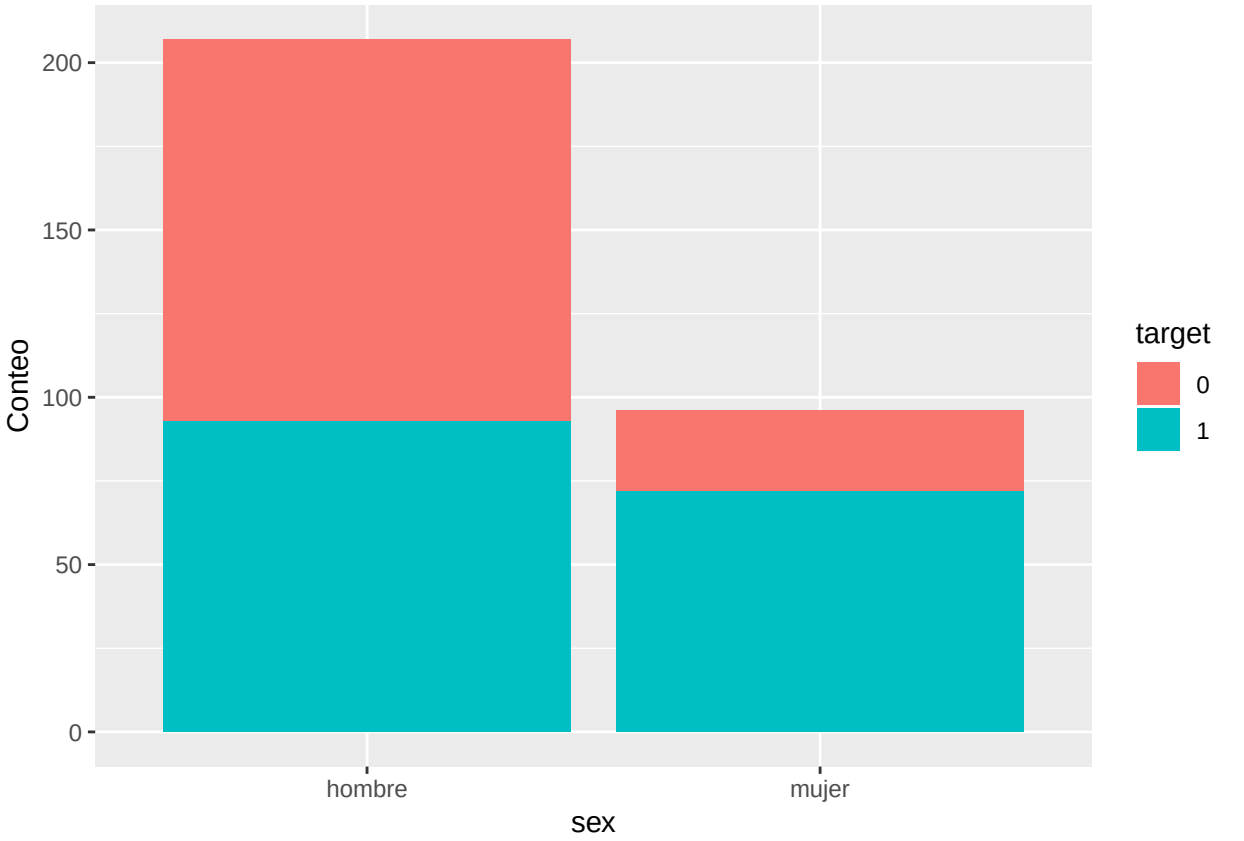
Como resultado, el algoritmo de árboles de clasificación puede predecir en un **80%** las enfermedades del corazón sobre el conjunto de pacientes de nuestro dataset.

Representación de los resultados a partir de tablas y gráficas.

Hemos observado a través de los diferentes modelos construidos, que tan efectivo es el dataset para precedir enfermedades cardiacas, es interesante analizar la relación entre las variables descriptoras con el campo objetivo `targe` con el fin de obtener cierta información acerca de los factores predominantes al momento de que un paciente contrae una enfermedad cardiaca.

Por ejemplo, analizamos el porcentaje de personas que con enfermedades del corazón en relación con el sexo.

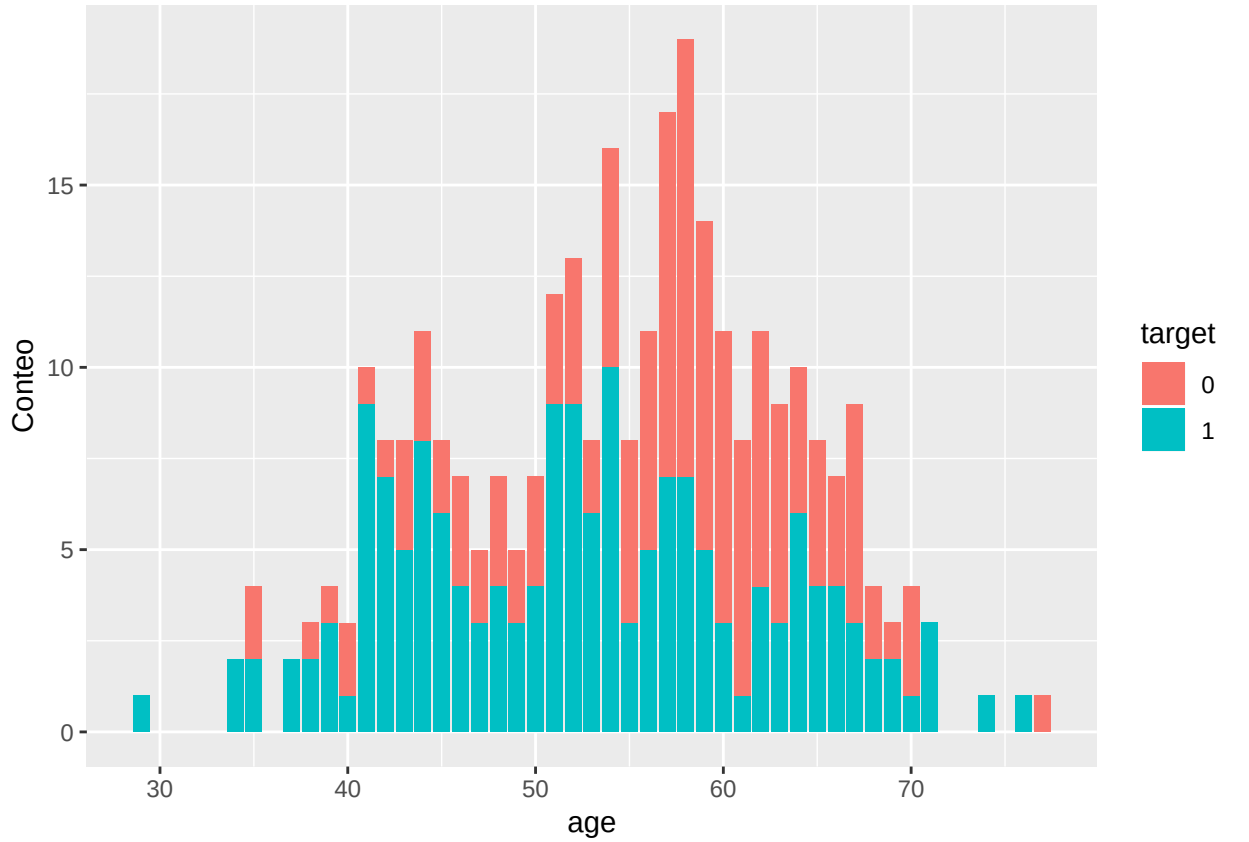
```
filas=dim(heart)[1]
ggplot(data = heart[1:filas,],aes(x=sex,fill=target))+geom_bar()+ylab("Conteo")
```



Se puede observar una proporción case de 2 a 1 entre mujeres y hombres, pero las mujeres son las que más presentan enfermedades del corazón.

Un análisis interesante también es el de revisar las enfermedades del corazón de los pacientes en relación con la edad de los mismos.

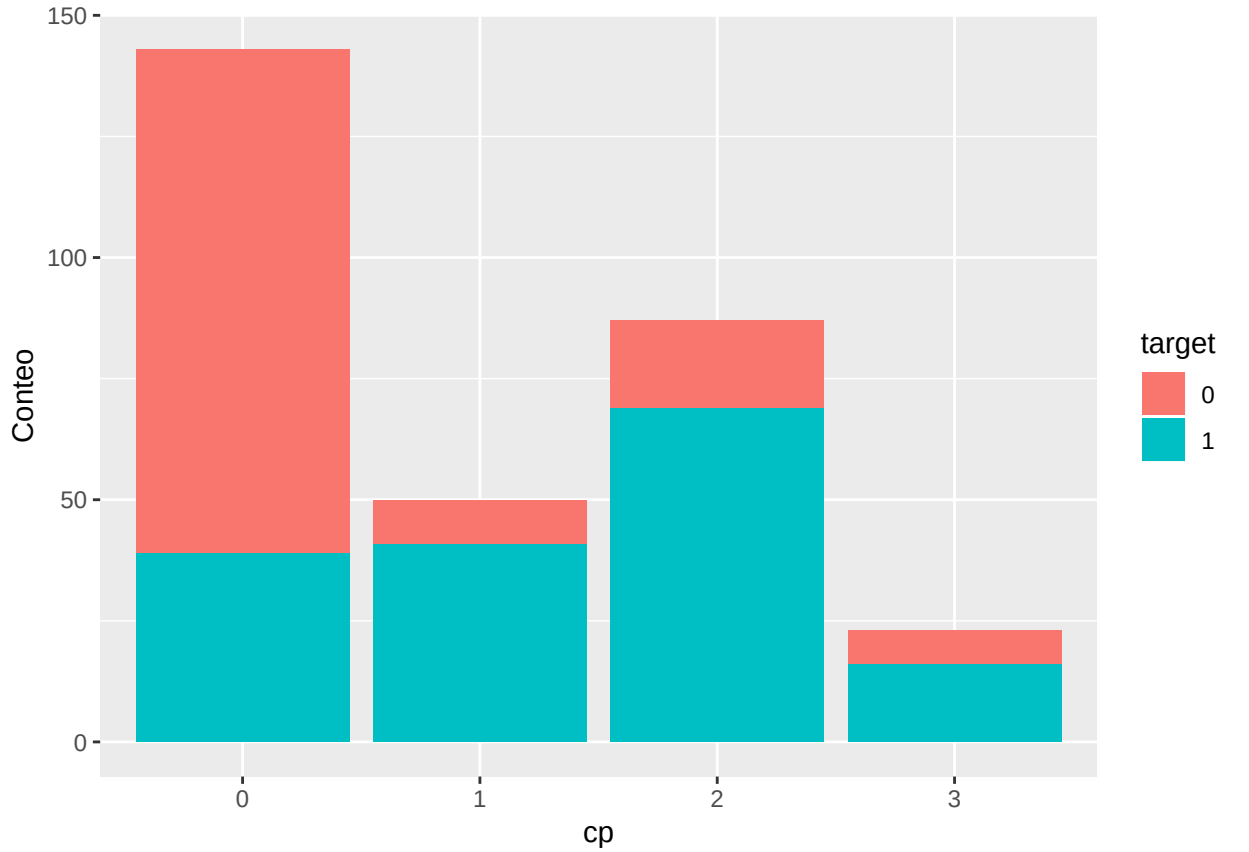
```
ggplot(data = heart[1:filas,],aes(x=age,fill=target))+geom_bar()+ylab("Conteo")
```



Existe cierta tendencia en que las personas de entre 50 y 60 años son las quemás sufren enfermedades del corazón.

La variable `cp` recoje el tipo de dolor en pecho, podemos buscar una relación entre estos dolores y una enfermedad en el corazón.

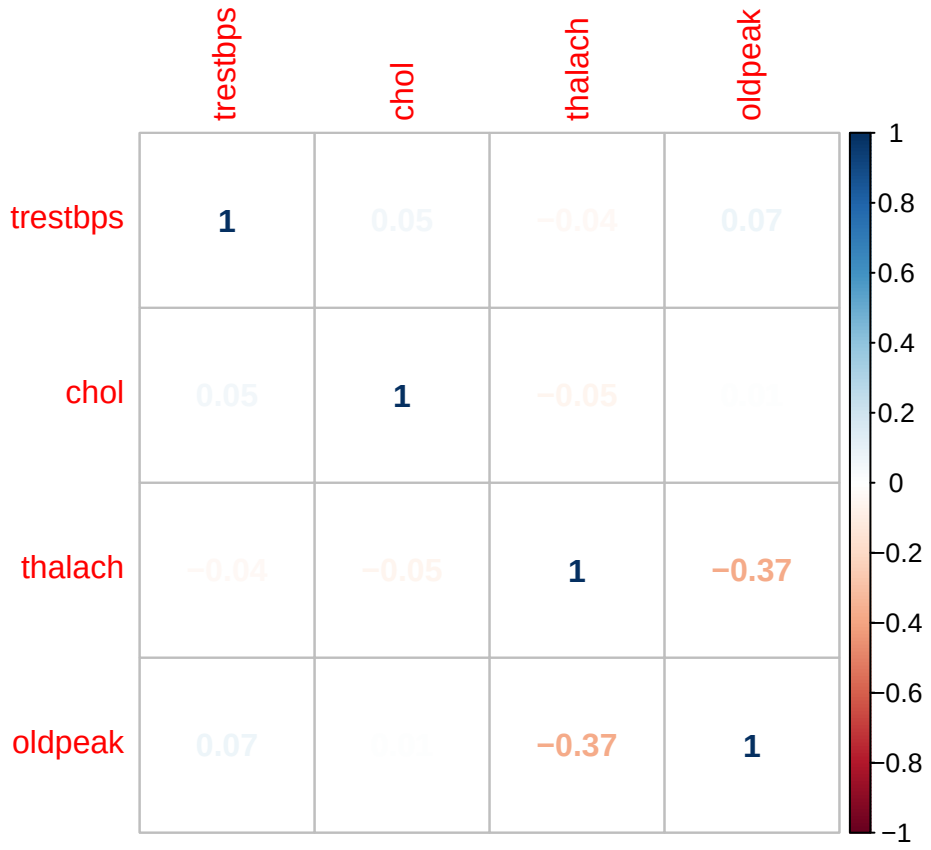
```
ggplot(data = heart[1:filas,],aes(x=cp,fill=target))+geom_bar()+ylab("Conteo")
```



En los casos en donde el dolor va desde moderado hasta intenso existe uns muy fuerte probabilidad de que el paciente tenga una enfermedad cardíaca.

Un análisis demás interesante es el encontrar la correlación entre las variables y así poder determinar que similitudes existen. Es este análisis revisaremos las variables continuas del dataset.

```
M <- cor(subset(heart, select = c(trestbps,chol,thalach,oldpeak)))
corrplot(M, method="number")
```



no existe una relación muy fuerte entre las variables continuas del dataset.

Conclusiones

Entre las principales conclusiones que podemos obtener:

1. Los atributos que conforman el dataset permiten generar modelos de clasificación que permiten predecir con un alto índice de precisión si un paciente es propenso a sufrir alguna enfermedad del corazón.
2. Existen algunas variables como `sex` y `age` que nos permiten concluir su influencia en las presencia de enfermedades cardiacas.
3. El dataset cumple su cometido de servir como insumo para el desarrollo de modelos que permitan oredecir enfermedades cardiacas según ciertos atributos, aunque debería considerarse realizar ciertas tareas de procesamiento previas como la normalización de variables y poda en los árboles de decisión.

Contribuciones

Contribuciones	Firma
Investigación previa	DC, CH
Redacción de las respuestas	DC, CH
desarrollo de código	DC, CH