Tipología y ciclo de vida de los datos - PRA-2

Alberto Perez Gant y Patricia García Menendez

12 de January, 2023

Contents

1 Descripción del dataset

En el presente trabajo se empleará un dataset llamado "**Heart Attack Analysis & Prediction Dataset**" el cual ofrece 14 variables agrupadas en columnas por 303 filas. Se incluye la variable objetivo, la cual pretende determinar si existe riesgo de sufrir un ataque cardiaco o no.

En primer lugar, se realiza la carga del archivo CSV "heart.csv" ubicado en el mismo directorio que el presente fichero .Rmd.

```
df <- read.csv("../dataset/heart.csv", stringsAsFactors = TRUE)
head(df)
```

```
##   age sex cp trtbps chol fbs restecg thalachh exng oldpeak slp caa thall output
## 1  63   1  3    145  233   1       0      150    0     2.3   0   0     1      1
## 2  37   1  2    130  250   0       1      187    0     3.5   0   0     2      1
## 3  41   0  1    130  204   0       0      172    0     1.4   2   0     2      1
## 4  56   1  1    120  236   0       1      178    0     0.8   2   0     2      1
## 5  57   0  0    120  354   0       1      163    1     0.6   2   0     2      1
## 6  57   1  0    140  192   0       1      148    0     0.4   1   0     1      1
```

Una vez se ha importado el dataset y almacenado en la variable que contiene los datos del ejercicio, se procede a su exploración. En primer lugar, aunque se ha realizado una visualización de la cabecera del dataframe mediante el método **head()**, se examina que tipo de datos contiene con **str()**.

```
str(df)
```

```
## '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 ...
##  $ trtbps  : 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 ...
##  $ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng    : 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 ...
##  $ slp     : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ caa     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thall   : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ output  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Las variables encontradas son:

- Age : Edad del paciente
- Sex : Sexo del paciente
  - 0 = mujer
  - 1 = hombre
- exang: angina inducida por el ejercicio
  - 1 = sí
  - 0 = no
- caa: Número de arterias principales (0-3)
- cp : Dolor en el pecho tipo de dolor en el pecho
  - Valor 1: angina típica
  - Valor 2: angina atípica
  - Valor 3: dolor no anginoso
  - Valor 4: asintomático
- trtbps : presión arterial en reposo (en mm Hg)
- chol : colesterol en mg/dl obtenido a través del sensor de IMC
- fbs : (azúcar en sangre en ayunas > 120 mg/dl)
  - 0 = falso
  - 1 = verdadero
- rest_ecg : resultados electrocardiográficos en reposo
  - Value 0: normal
  - Value 1: tener anomalía de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST de $> 0{,}05$ mV)
  - Value 2: mostrar hipertrofia ventricular izquierda probable o definida según el criterio de Estes
- thalach : frecuencia cardíaca máxima alcanzada

- oldpeak: pico anterior
- slp: Slope
- thall: thall rate
- target :
  - 0= Menos riesgo de sufrir un ataque cardiaco
  - 1= Mas riesgo de sufrir un ataque cardiaco

Como se puede observar, la mayoría de las variables son de tipo numérico, por lo que será necesario convertirlas a un tipo de dato acorde al objetivo de la misma. Se emplea ahora la función **summary()** para hacer un resumen estadísitico de las variables del dataset.

```
summary(df)
```

```
##       age             sex               cp            trtbps
##  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          thalachh
##  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
##       exng           oldpeak          slp              caa
##  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
##      thall           output
##  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 observa como algunas de las variables arrojan unos estadísticos sin sentido como en el caso de la variable sexo o rest_ecg, lo que indica que su formato puede que no sea el más apropiado para su representación y análisis.

2 Integración y selección

Como se ha comentado en el apartado anterior se deben realizar cambios en los formatos de algunas de las variables. A continuación, se muestran las conversiones de las variables para que contengan un formato favorable a un posterior análisis.

```
#Conversión de variables categóricas a factor
df$sex <- as.factor(ifelse(df$sex == 0, "Female", "Male"))
df$exng <- as.factor(ifelse(df$exng == 0, "FALSE", "TRUE"))
df$cp <- as.factor(df$cp)
df$fbs <- as.factor(ifelse(df$fbs == 0, "FALSE", "TRUE"))
df$restecg <- as.factor(df$restecg)
df$output <- as.factor(ifelse(df$output == 0, "FALSE", "TRUE"))

#Conversión de variables categóricas a numeric
df$age <- as.numeric(df$age)
df$trtbps <- as.numeric(df$trtbps)
df$chol <- as.numeric(df$chol)
df$thalachh <- as.numeric(df$thalachh)
df$oldpeak <- as.numeric(df$oldpeak)
df$slp <- as.numeric(df$slp)
df$caa <- as.numeric(df$caa)
df$thall <- as.numeric(df$thall)
df$output_num <- as.numeric(ifelse(df$output == "FALSE", 0,1))
head(df)
```

```
##   age    sex cp trtbps chol   fbs restecg thalachh  exng oldpeak slp caa thall
## 1  63   Male  3    145  233  TRUE       0      150 FALSE     2.3   0   0     1
## 2  37   Male  2    130  250 FALSE       1      187 FALSE     3.5   0   0     2
## 3  41 Female  1    130  204 FALSE       0      172 FALSE     1.4   2   0     2
## 4  56   Male  1    120  236 FALSE       1      178 FALSE     0.8   2   0     2
## 5  57 Female  0    120  354 FALSE       1      163  TRUE     0.6   2   0     2
## 6  57   Male  0    140  192 FALSE       1      148 FALSE     0.4   1   0     1
##   output output_num
## 1   TRUE          1
## 2   TRUE          1
## 3   TRUE          1
## 4   TRUE          1
## 5   TRUE          1
## 6   TRUE          1
```

En primer lugar, se ha convertido la variable *sex* que era binaria a una categórica de dos niveles (Female/ Male). Las variables binarias *exng*,*fbs* y *output* se han convertido a categóricas de tipo TRUE/FALSE. Mientras que las variables *cp* y *restecg* cambian de tipo int a tipo factor sin modificación en los valores que pueden tomar. Por otro lado, las varibles de tipo int como *age*, *trtbps*, *chol*, *thalachh*, *oldpeak*, *slp*, *caa* y *thall* pasan a ser de tipo numérico.

A continución, se vuelve a mostrar la tipología y el resumen estadístico de las variables del dataset después de su modificación.

```
str(df)
```

```
## 'data.frame':    303 obs. of  15 variables:
##  $ age       : num  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex       : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 1 2 2 2 ...
##  $ cp        : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ trtbps    : num  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol      : num  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs       : Factor w/ 2 levels "FALSE","TRUE": 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 ...
##  $ thalachh  : num  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng      : Factor w/ 2 levels "FALSE","TRUE": 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 ...
##  $ slp       : num  0 0 2 2 2 1 1 2 2 2 ...
##  $ caa       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ thall     : num  1 2 2 2 2 1 2 3 3 2 ...
##  $ output    : Factor w/ 2 levels "FALSE","TRUE": 2 2 2 2 2 2 2 2 2 2 ...
##  $ output_num: num  1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(df)
```

```
##       age            sex      cp          trtbps           chol
##  Min.   :29.00   Female: 96   0:143   Min.   : 94.0   Min.   :126.0
##  1st Qu.:47.50   Male  :207   1: 50   1st Qu.:120.0   1st Qu.:211.0
##  Median :55.00                2: 87   Median :130.0   Median :240.0
##  Mean   :54.37                3: 23   Mean   :131.6   Mean   :246.3
##  3rd Qu.:61.00                        3rd Qu.:140.0   3rd Qu.:274.5
##  Max.   :77.00                        Max.   :200.0   Max.   :564.0
##     fbs      restecg    thalachh        exng        oldpeak          slp
##  FALSE:258   0:147   Min.   : 71.0   FALSE:204   Min.   :0.00   Min.   :0.000
##  TRUE : 45   1:152   1st Qu.:133.5   TRUE : 99   1st Qu.:0.00   1st Qu.:1.000
##              2:  4   Median :153.0               Median :0.80   Median :1.000
##                      Mean   :149.6               Mean   :1.04   Mean   :1.399
##                      3rd Qu.:166.0               3rd Qu.:1.60   3rd Qu.:2.000
##                      Max.   :202.0               Max.   :6.20   Max.   :2.000
##       caa             thall         output      output_num
##  Min.   :0.0000   Min.   :0.000   FALSE:138   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:2.000   TRUE :165   1st Qu.:0.0000
##  Median :0.0000   Median :2.000               Median :1.0000
##  Mean   :0.7294   Mean   :2.314               Mean   :0.5446
##  3rd Qu.:1.0000   3rd Qu.:3.000               3rd Qu.:1.0000
##  Max.   :4.0000   Max.   :3.000               Max.   :1.0000
```

Al realizar un resumen posterior de las variables modificadas, se observa como aquellas que antes tenían un formato entero, se ha conseguido que ahora se pueda ofrecer un conteo sobre las que presentan características categóricas.

3 Limpieza de datos

En el presente apartado se realizará una limpieza de los datos para determinar si existen elementos vacíos o valores anómalos.

3.1 Gestión de elementos vacíos

En primer lugar, se va a comprobar si existen variables que contengan algún registro vacío. Para ello se emplearán dos métodos concatenados. Por un lado, *is.na()* que indica si hay algún nulo y por otro *any()* que devuelve TRUE si hay algún resgitro como TRUE o FALSE en caso contrario.

```
any(is.na(df))
```

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

Como se puede obervar, no existe ningún elemento vacío en el conjunto del dataset. Por otro lado, sí existen ceros, ya que algunas variables pueden tomar como valor el 0. Algunas de variables donde puede suceder esto son:

- cp
- restecg

- slp
- caa
- oldpeak
- thall

3.2 Tratamiento de outliers

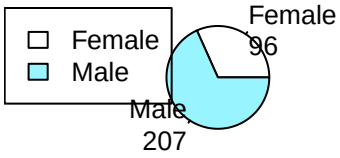
Para realizar el tratamiento de outliers en primer lugar se tienen que identificar. Para ello se va a hacer una representación de las variables. Cabe destacar que las variables de tipo factor dado que son categóricas no puede darse este problema por lo que los outliers se estudiarán en aquellas variables de tipo numérico.

En primer lugar, se va a ver la distribución de las variables. Para ello se realizarán diagramas de tarta en el caso de las categóricas e histogramas en el caso de las numéricas.

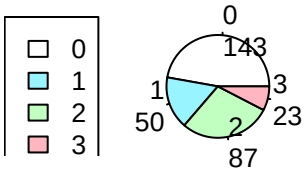
```
table_sex <- table(df$sex)
lbls_sex <- paste(names(table_sex), "\n", table_sex, sep="")
table_cp <- table(df$cp)
lbls_cp <- paste(names(table_cp), "\n", table_cp, sep="")
table_fbs <- table(df$fbs)
lbls_fbs <- paste(names(table_fbs), "\n", table_fbs, sep="")
table_restecg <- table(df$restecg)
lbls_restecg <- paste(names(table_restecg), "\n", table_restecg, sep="")

par(mfrow=c(2,2))
color <- c("white","cadetblue1","darkseagreen1", "lightpink")
pie(table_sex, labels = lbls_sex, main="Num. de mujeres y hombres", col = color)
legend("topleft", c("Female","Male"),fill =  color)
pie(table_cp, labels = lbls_cp, main="Num. pac.según dolor de pecho",col =color)
legend("topleft", c("0","1","2","3"),fill =  color)
pie(table_fbs, labels = lbls_fbs, main="Num. pac. con azúcar alto", col = color)
legend("bottomleft", c("FALSE","TRUE"),fill =  color)
pie(table_restecg, labels = lbls_restecg, main="Num. pac. según resultados ECG",
    col = color)
legend("topleft", c("0","1","2"),fill =  color)
```

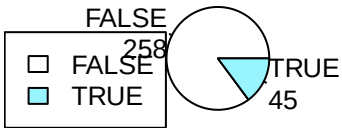
Num. de mujeres y hombres



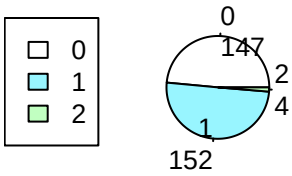
Num. pac.según dolor de pecho



Num. pac. con azúcar alto

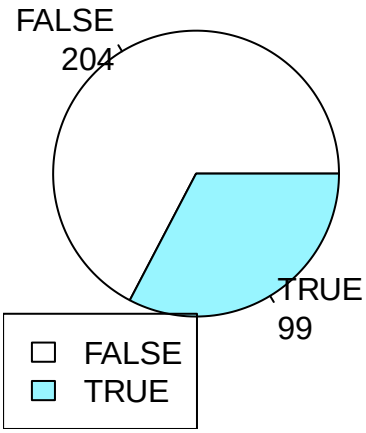


Num. pac. según resultados ECG

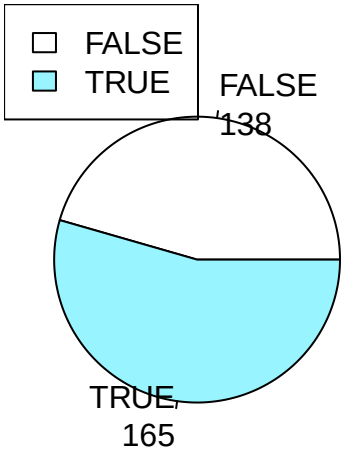



```
table_exng <- table(df$exng)
lbls_exng <- paste(names(table_exng), "\n", table_exng, sep="")
table_output <- table(df$output)
lbls_output <- paste(names(table_output), "\n", table_output, sep="")
par(mfrow=c(1,2))
pie(table_exng, labels = lbls_exng, main="Num. pac. con angina ejercicio",
    col = color)
legend("bottomleft", c("FALSE","TRUE"),fill =  color)
pie(table_output, labels = lbls_output, main="Num. pac. con riesgo alto",
    col = color)
legend("topleft", c("FALSE","TRUE"),fill =  color)
```

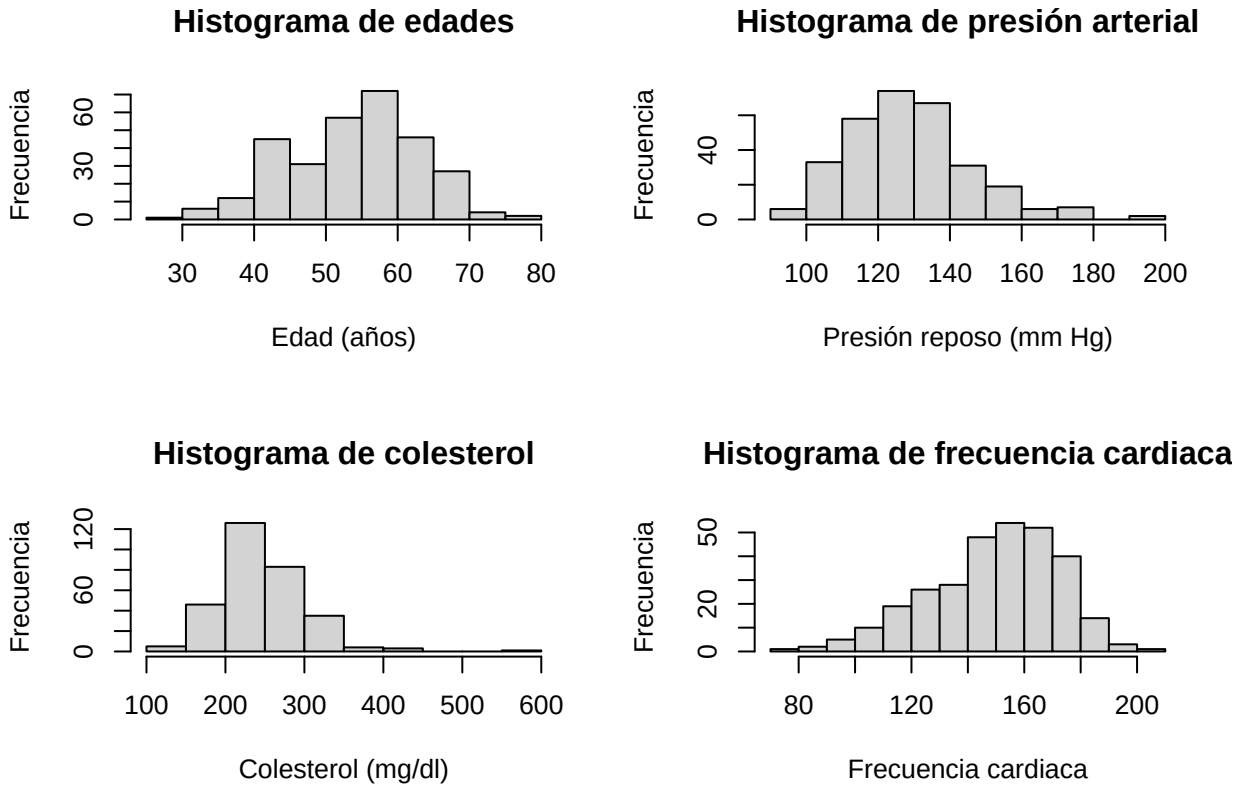
Num. pac. con angina ejercicio



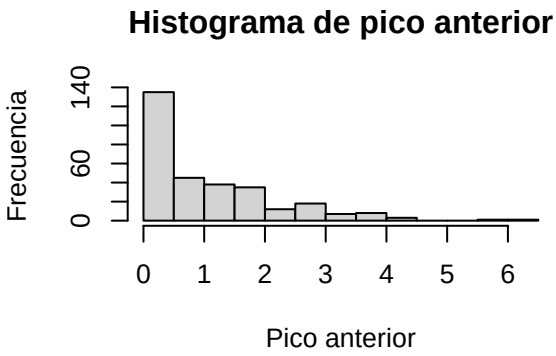
Num. pac. con riesgo alto

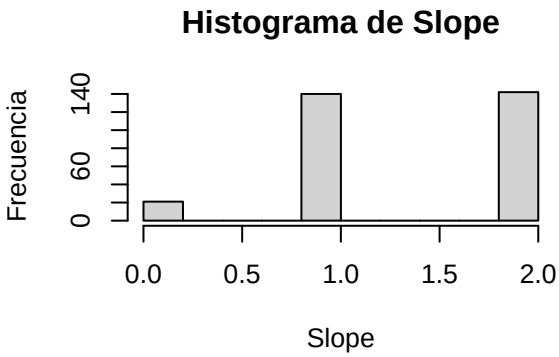


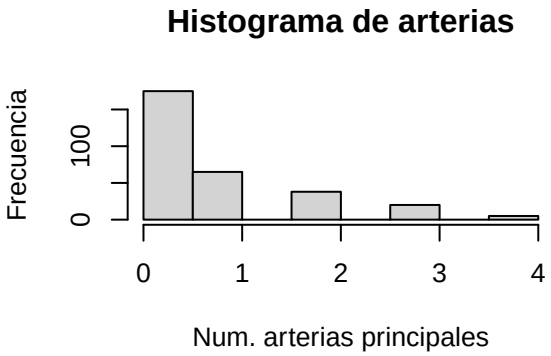
```
par(mfrow=c(2,2))
hist(df$age, main="Histograma de edades", xlab="Edad (años)", ylab="Frecuencia")
hist(df$trtbps, main="Histograma de presión arterial",
     xlab="Presión reposo (mm Hg)", ylab="Frecuencia")
hist(df$chol, main="Histograma de colesterol",
     xlab="Colesterol (mg/dl)", ylab="Frecuencia")
hist(df$thalachh, main="Histograma de frecuencia cardiaca",
     xlab="Frecuencia cardiaca", ylab="Frecuencia")
```

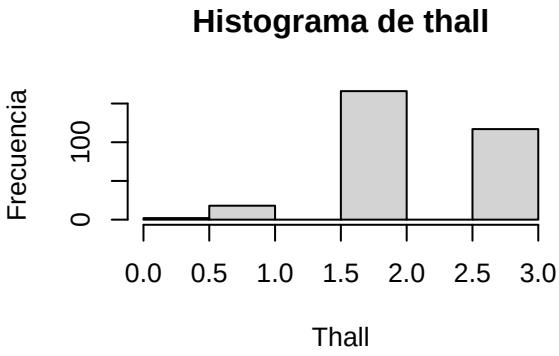



```
par(mfrow=c(2,2))
hist(df$oldpeak, main="Histograma de pico anterior", xlab="Pico anterior",
     ylab="Frecuencia")
hist(df$slp, main="Histograma de Slope", xlab="Slope", ylab="Frecuencia")
hist(df$caa, main="Histograma de arterias", xlab="Num. arterias principales",
     ylab="Frecuencia")
hist(df$thall, main="Histograma de thall", xlab="Thall", ylab="Frecuencia")
```



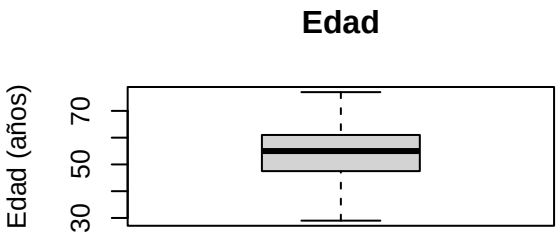


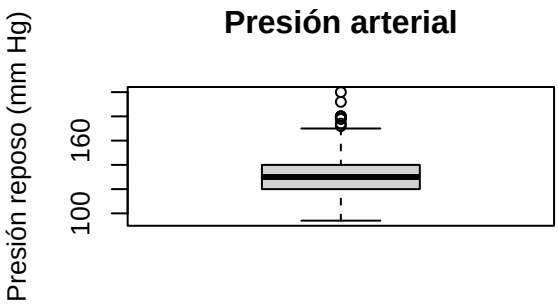


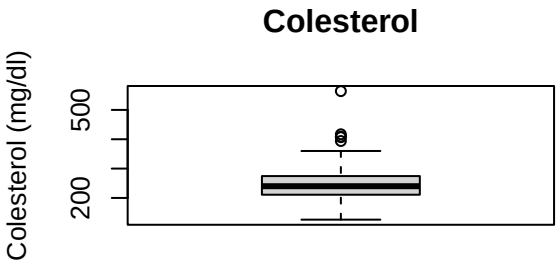


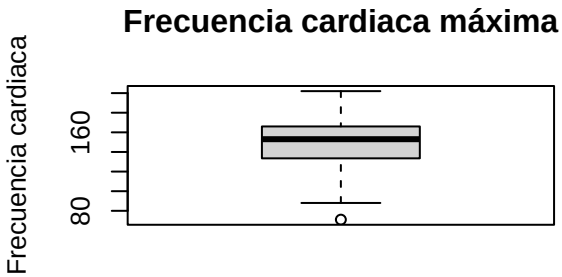
Viendo los histogramas se puede ver como se distribuyen las variables pero para ver los valores extremos (outliers) se van a realizar los diagramas de cajas y bigotes **boxplot()**.

```
par(mfrow=c(2,2))
boxplot(df$age, main="Edad", ylab="Edad (años)")
boxplot(df$trtbps, main="Presión arterial", ylab="Presión reposo (mm Hg)")
boxplot(df$chol, main="Colesterol", ylab="Colesterol (mg/dl)")
boxplot(df$thalachh, main="Frecuencia cardiaca máxima",
        ylab="Frecuencia cardiaca")
```

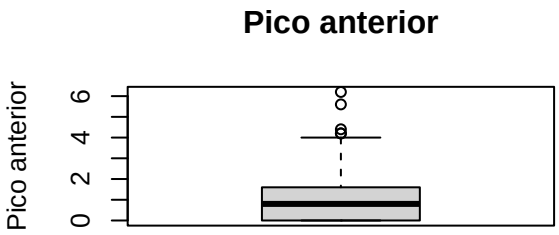


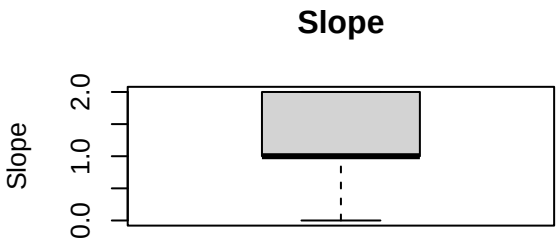


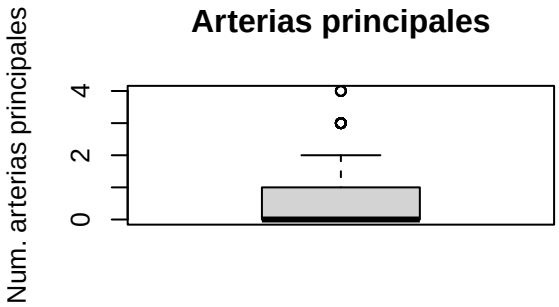


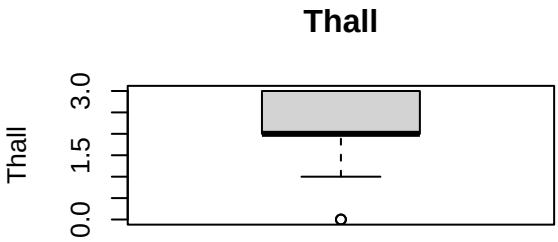



```
par(mfrow=c(2,2))
boxplot(df$oldpeak, main="Pico anterior", ylab="Pico anterior")
boxplot(df$slp, main="Slope", ylab="Slope")
boxplot(df$caa, main="Arterias principales", ylab="Num. arterias principales")
boxplot(df$thall, main="Thall", ylab="Thall")
```









A continuación, se va a analizar cada varaible:

- Edad: como se puede observar en la gráfica no se tiene ningún valor extremo.

- Presion arterial: se tienen varios pacientes cuya presión arterial está por encima del límite superior (170 mm Hg) por lo que se considerarían outliers.
- Colesterol: se puede ver que hay pacientes que están por encima del límite superior de 350 mg/dl por lo que también se considerarían outliers.
- Frecuencia cardiaca:se puede observa que hay un paciente que tiene la frecuencia cardiaca máxima por debajo de 90 que es el límite inferior. En este caso se trataría de un outlier inferior.
- Pico anterior: en este caso se ve que hay outliers más altos al valor límite superior (4).
- Slope: en esta variable no se presentan valores extremos.
- Número de arterias principales: se ve que lo normal es que sean 2 o menos vasos sanguíneos afectados pero se pueden encontrar algunos registros en los que se da que este valor es superior a 2.
- Thall: en este caso se observa que hay un caso en el que la variable toma un valor atípico con respecto al límite inferior lo que implica la presencia de un outlier inferior.

Para poder realizar un estudio que sea lo más preciso posible se han marcado todos aquellos datos que puedan influir de manera directa en la interpretación del resultado o en la aplicación de modelos de predicción, es por ello que se procede a marcar todos aquellos registros considerados como outliers para tenerlos identificados y realizar un estudio aparte si fuera necesario.

```
out <- boxplot.stats(df$trtbps)$out
out_trtbps <- which(df$trtbps %in% c(out))
out_trtbps
```

```
## [1]   9 102 111 204 224 242 249 261 267
```

```
out <- boxplot.stats(df$chol)$out
out_chol <- which(df$chol %in% c(out))
out_chol
```

```
## [1]  29  86  97 221 247
```

```
out <- boxplot.stats(df$thalachh)$out
out_thalachh <- which(df$thalachh %in% c(out))
out_thalachh
```

```
## [1] 273
```

```
out <- boxplot.stats(df$oldpeak)$out
out_oldpeak <- which(df$oldpeak %in% c(out))
out_oldpeak
```

```
## [1] 102 205 222 251 292
```

```
out <- boxplot.stats(df$caa)$out
out_caa <- which(df$caa %in% c(out))
out_caa
```

```
##  [1]  53  93  98 100 159 164 165 166 182 192 205 209 218 221 232 235 239 248 250
## [20] 251 252 253 256 268 292
```

```
out <- boxplot.stats(df$thall)$out
out_thall <- which(df$thall %in% c(out))
out_thall
```

```
## [1]  49 282
```

```
out_pos <- sort(unique(c(out_trtbps, out_chol, out_thalachh, out_oldpeak,
                         out_caa, out_thall)))
out_pos
```

```
##  [1]   9  29  49  53  86  93  97  98 100 102 111 159 164 165 166 182 192 204 205
```

```
## [20] 209 218 221 222 224 232 235 239 242 247 248 249 250 251 252 253 256 261 267
## [39] 268 273 282 292
```

```
df[out_pos, "ind_out"] = 1
df$ind_out[is.na(df$ind_out)] <- 0
df$ind_out <- as.factor(df$ind_out)
```

El proceso anterior permite crear una variable más en el dataset que indica si el registro es un outlier debido a cualquiera de sus variables. Se indica con 1 aquellos registro que se consideran outlier y con 0 los que no. Esto permitirá identificarlos de manera más sencilla a la hora de hacer el análisis.

4 Análisis de datos

4.1 Selección de los grupos de datos que se quieren analizar y comparar

Como se ha comentado en el apartado anterior se va a analizar el dataset completo, se han marcado todos aquellos registros que se consideran valores extremos en alguna de las variables de estudio y si fuera necesario tras analizar los resultados se eliminarán del conjunto, pero a priori parece que esos outliers pueden ser de utilidad tanto en el estudio y análisis como en la generación de modelos predictivos.

4.2 Comprobación de normalidad y homogeneidad de la varianza

A continuación, se emplea la prueba de normalidad de Anderson-Darling para las variables cuantitativas del dataset. Para ello, se considera un nivel de significación de 0.05.

```
if (!require("nortest")) install.packages("nortest")
```

```
## Loading required package: nortest
```

```
library(nortest)

alpha = 0.05
col.names = colnames(df)

for (i in 1:ncol(df)) {

  if (i == 1) cat("Variables que no siguen una distribución normal:\n")

  if (is.integer(df[,i]) | is.numeric(df[,i])) {
    p_val = ad.test(df[,i])$p.value

    if (p_val < alpha) {
      cat(col.names[i])

    # Format output
    if (i < ncol(df) - 1) cat(", ")
    if (i %% 3 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal:
## age, trtbps, chol, thalachh, oldpeak, slp, caa,
## thall, output_num
```

Como se puede observar, las variables que no siguen una distribución normal con un nivel de significación de 0.05 son:

- age
- trtbps
- chol
- thalachh
- oldpeak
- slp
- caa
- thall

Esto indica que ninguna de las variables presenta normalidad para un nivel de significacia de 0.05.

Posteriormente, para estudiar la homogeneidad de varianzas, se aplica un test de Fligner-Killen. En este caso, se aplica de acuerdo a la edad y el colesterol de los pacientes. La hipótesis nula consiste en que ambas varianzas sean iguales.

```
fligner.test(age ~ chol, data=df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  age by chol
## Fligner-Killeen:med chi-squared = 153.6, df = 151, p-value = 0.4258
```

Dado que se obtiene un p-valor superior a 0.05, se puede afirmar que las varianzas de ambas variables son homogéneas.

Además, también es interesante comprobar como ofectan otras variables frente a la edad como oldpeak.

```
fligner.test(age ~ oldpeak, data=df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  age by oldpeak
## Fligner-Killeen:med chi-squared = 42.372, df = 39, p-value = 0.3277
```

Nuevamente, se observa como ambas varianzas son homogéneas.

Otras variables que interesan comparar con la edad son trtbps y thalachh.

```
fligner.test(age ~ trtbps, data=df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  age by trtbps
## Fligner-Killeen:med chi-squared = 59.6, df = 48, p-value = 0.1216
```

```
fligner.test(age ~ thalachh, data=df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  age by thalachh
## Fligner-Killeen:med chi-squared = 84.59, df = 90, p-value = 0.6412
```

Como se puede apreciar, todas las combinaciones analizadas son homogéneas.

4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos

Cálculo y análisis de correlaciones entre variables:

En primer lugar se va a comprobar la correlación que hay entre las variables del dataset. Para ello, se genera la matriz de correlaciones para todas las variables numéricas.

```
df_num <- df[c("age", "trtbps", "chol", "thalachh", "oldpeak", "slp", "caa",
               "thall", "output_num")]
correlaciones <- round(cor(df_num), 2)
correlaciones
```

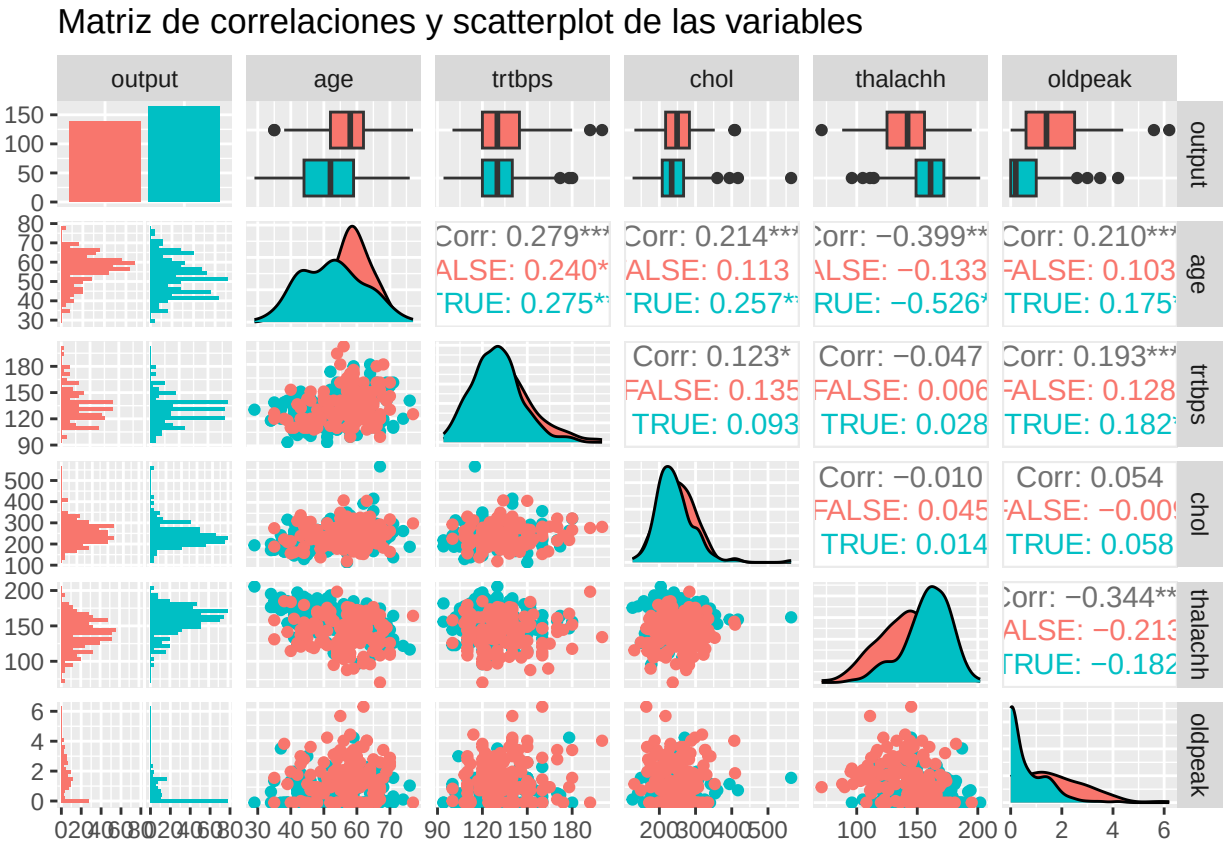
```
##              age trtbps  chol thalachh oldpeak   slp   caa thall output_num
## age         1.00   0.28  0.21    -0.40    0.21 -0.17  0.28  0.07      -0.23
## trtbps      0.28   1.00  0.12    -0.05    0.19 -0.12  0.10  0.06      -0.14
## chol        0.21   0.12  1.00    -0.01    0.05  0.00  0.07  0.10      -0.09
## thalachh   -0.40  -0.05 -0.01     1.00   -0.34  0.39 -0.21 -0.10       0.42
## oldpeak     0.21   0.19  0.05    -0.34    1.00 -0.58  0.22  0.21      -0.43
## slp        -0.17  -0.12  0.00     0.39   -0.58  1.00 -0.08 -0.10       0.35
## caa         0.28   0.10  0.07    -0.21    0.22 -0.08  1.00  0.15      -0.39
## thall       0.07   0.06  0.10    -0.10    0.21 -0.10  0.15  1.00      -0.34
## output_num -0.23  -0.14 -0.09     0.42   -0.43  0.35 -0.39 -0.34       1.00
```

Como se puede observar en la matriz anterior no existe mucha correlación entre las variables. Se puede observar que las variables que más correlación tienen con **output_num** son **thalachh** y **oldpeak**

En este caso, estudiaremos cuales de las variables cuantitativas influyen más en unas sobre otras. Para ello emplearemos la matriz generada por ggpairs, donde se indica el valor de correlación entre las variables consideradas.

```
var = c("output", "age", "trtbps", "chol", "thalachh", "oldpeak")
ggpairs(df, columns=var, aes(color=output)) +
  labs(title="Matriz de correlaciones y scatterplot de las variables")
```

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



Del anterior gráfico, se observa como no hay una gran correlación entre la mayoría de las varibales. No obstante, se observa como la edad es una de las que más relaciones contiene. Por otro lado, sobre la variable output, se observa como existe una media considerablemente mayor de personas con riesgo conforme aumenta el valor. Sucede lo mismo con el colesterol, la frecuencia máxima alcanzada (a la inversa) y oldpeak.

Contraste de hipótesis

Se va a realizar ahora un contraste de hipótesis sobre las muestras del dataset para determinar si el sexo y el nivel de azúcar en sangre influyen en el riesgo de sufrir un ataque al corazón.

Se empieza analizando la variable **sex**, para ello se generan dos muestras una para el sexo femenino y otra para el masculino:

```
df_female <- df[df$sex == "Female",]$output_num
df_male <- df[df$sex == "Male",]$output_num
```

Tomando $\alpha = 0.05$, se plantean las siguientes hipótesis:

```
  - H0: Los hombres tienen mayor número de ataques cardíacos que las mujeres.
  - H1: Los hombres tienen menor número de ataques cardíacos que las mujeres.
```

```
t.test(df_female,df_male, alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  df_female and df_male
## t = 5.3372, df = 209.95, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
```

```
##       -Inf 0.3938151
## sample estimates:
## mean of x mean of y
## 0.7500000 0.4492754
```

Como se puede observar el p-value está por encima de 0.05 por lo que se acepta la hipótesis nula lo que indica que los hombres tienen mayor número de ataques cardíacos que las mujeres con un nivel de confianza del 95% en el intervalo de confianza.

Se va a analizar ahora el nivel de azúcar en sangre. Para ello se analiza la variable **fbs**, para ello se genera como en el caso anterior dos grupos uno con el azúcar por encima de 120 mg/dl y otro con los que están por debajo:

```
df_fbs_true <- df[df$fbs == "TRUE",]$output_num
df_fbs_false <- df[df$fbs == "FALSE",]$output_num
```

Tomando $\alpha = 0.05$, se plantean las siguientes hipótesis:

- H0: Los pacientes con mayor azúcar tienen mayor número de ataques cardíacos.
- H1: Los paciente con mayor azúcar no tienen mayor número de ataques cardíacos.

```
t.test(df_fbs_true,df_fbs_false)
```

```
##
##  Welch Two Sample t-test
##
## data:  df_fbs_true and df_fbs_false
## t = -0.48193, df = 59.891, p-value = 0.6316
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2023025  0.1237496
## sample estimates:
## mean of x mean of y
## 0.5111111 0.5503876
```

Como se puede observar, en este caso, el p-value también está por encima de 0.05 por lo que se acepta la hipótesis nula, lo que dice que los pacientes con azúcar por encima de los 120 mg/dl tienen mayor número de ataques cardíacos que los que tienen el azúcar en sangre más bajo con un nivel de confianza del 95%.

Modelo de regresión lineal:

Tras realizar el estudio de las correlaciones y los contrastes de hipótesis se va ahora a realizar un modelo de regresió lineal con el objetivo de predecir si el paciente en cuestión tiene o no riesgo de sufrir un ataque cardíaco. Para ello se va a generan varios modelos en función de la matriz de correlación obtenida en uno de los apartados anteriores.

Se seleccionan los regresores cuantitativos con mayor coeficiente de correlación con respecto a la variable *output*, los regresores cualitativos y la variable a predecir:

```
# Regresores cuantitativos:
edad <- df$age
presion <- df$trtbps
colesterol <- df$chol
frec_cardiaca <- df$thalachh
pico <- df$oldpeak
slope <- df$slp
arterias <- df$caa
thall <- df$thall
```

```r
# Regresores cualitativos.
sexo <- df$sex
cp <- df$cp
azucar <- df$fbs
res_ECG <- df$restecg
angina <- df$exng

#Variable a predecir:
target <- df$output_num
```

Se generan los modelos a partir de los regresores seleccionados:

```r
# Modelo 1 con todas las variables:
modelo1 <- lm(target ~ edad +  colesterol + frec_cardiaca + pico + slope +
                arterias + thall + sexo + cp + azucar + res_ECG + angina,
              data = df)
# Modelo 2:
modelo2 <- lm(target ~ edad +  frec_cardiaca +  pico , data = df)

# Modelo 3:
modelo3 <- lm(target ~ edad +   frec_cardiaca + pico +  sexo + cp + azucar
              + res_ECG , data = df)

res_modelos <- matrix(c("Modelo 1", summary(modelo1)$r.squared,
                        "Modelo 2",summary(modelo2)$r.squared,
                        "Modelo 3", summary(modelo3)$r.squared),
                      ncol = 2, byrow = TRUE)
res_modelos
```

```
##      [,1]       [,2]
## [1,] "Modelo 1" "0.52065032123398"
## [2,] "Modelo 2" "0.27171967061634"
## [3,] "Modelo 3" "0.443721950005421"
```

Como se puede ver los mejores resultados de predicción mediante los modelos de regresión se dan cuando se combinan todas las variables, aún así puede verse que el valor de R-squared es bastante bajo, esto puede deberse a la tipología del dataset, dado que es un problema más de clasificación que de regresión y por lo tanto se obtendrían mejores resultados aplicando modelos de clasificación como pueden ser KNN, árboles de decisión o clusters.

5 Resolución del problema

Como se ha ido explicando en la resolución de la práctica,el dataset recoge información sobre pacientes. Con la información obtenida se pretende dar respuesta a si un paciente tiene riesgo o no de sufrir un ataque cardíaco. Para ello se cuenta con información de la edad, sexo, presión sanguínea, frecuencia cardiaca… y también con la variable que se busca predecir,la cual indica si dicho paciente ha sufrido o no un ataque cardíaco. El dataset propuesto responde de manera bastante buena a la resolución del problema mediante los métodos utilizados, cabe destacar aún así que los resultados son bastante mejorables dado que se obtiene un R-squared de 0.52. Estos resultados podrían mejorar, como ya se ha comentado, aplicando algoritmos de clasificación en lugar de los de regresión o incluso la regresión logística.

Enlaces

Enlace al vídeo: https://drive.google.com/file/d/1lojUL9lUEX4xb_mV3VXn4ea7JiPgdZpF/view?usp=share_link

Enlace al repositorio de Github: https://github.com/AlbertoPerezGant/PRA2.git

Tabla de contribuciones:

Contribuciones	Firma
Investigación previa	Alberto Pérez, Patricia García
Redacción de las respuestas	Alberto Pérez, Patricia García
Desarrollo del código	Alberto Pérez, Patricia García
Participación en el vídeo	Alberto Pérez, Patricia García