UOC - Tipologia y ciclo de vida de los datos - PRA2

Limpieza y Preprocesado: Heart Attack Analysis & Prediction Dataset

Vanessa Moreno González, Manuel Ernesto Martínez Martín

28 de May 2023

Índice

1. Descripción del dataset

Este dataset trae dos ficheros `heart.csv` y `o2Saturation.csv` y **es importante porque proporciona información sobre factores relacionados con enfermedades cardíacas**, como edad, sexo, síntomas otros datos médicos. Ya que con el **se puede entender mejor la enfermedad y hacer un análisis para detectar cuando se puede estar en riesgo de ataque cardíaco**, sabiendo esto se pueden desarrollar modelos predictivos que tomen decisiones para ayudar a prevenir un ataque cardíaco.

El dataset es el propuesto en el enunciado de la práctica y se ha extraído de kaggle: **Heart Attack Analysis & Prediction Dataset**

Contenido del dataset

Las variables que tiene el dataset son: age, sex, cp, trtbps, chol, fbs, restecg, thalachh, exng, oldpeak, slp, caa, thall y output. Siendo `output` la variable objetivo. A continuación se detallan más en profundidad.

- **age**: Edad del paciente.
- **sex**: Género del paciente.
  - *0*: Femenino
  - *1*: Masculino
- **cp**: Tipo de dolor en el pecho.
  - *0*: Angina típica
  - *1*: Angina atípica
  - *2*: Dolor no anginal
  - *3*: Asintomático
- **trtbps**: Presión arterial en reposo (en mm Hg).
- **chol**: Colesterol en mg/dl medido mediante un sensor BMI.
- **fbs**: Nivel de azúcar en sangre en ayunas (> 120 mg/dl).
  - *1*: Verdadero
  - *0*: Falso
- **restecg**: Resultados electrocardiográficos en reposo.
  - *0*: Normal
  - *1*: Anormalidad con inversiones de onda ST-T y/o alteraciones del segmento ST > 0.05 mV
  - *2*: Hipertrofia ventricular izquierda
- **thalachh**: Ritmo cardíaco máximo alcanzado.
- **exng**: Angina inducida por ejercicio.
  - *1*: Sí
  - *0*: No
- **oldpeak**: Diferencia entre la depresión del segmento ST durante el ejercicio y durante el descanso en un electrocardiograma.
- **slp**: Pendiente del segmento ST durante el ejercicio en la prueba de esfuerzo.
  - *1*: Ascendente
  - *2*: Plana
  - *3*: Descendente
- **caa**: Número de vasos principales (0-3).
- **thall**: Talasemia, trastorno hereditario de la sangre caracterizado por un menor nivel de hemoglobina.
  - *0*: Ausencia

  - *1*: Talasemia normal
  - *2*: Talasemia fija defectuosa
  - *3*: Talasemia Reversible defectuosa

- **output**: Variable objetivo.
  - *0*: Menor probabilidad de ataque al corazón
  - *1*: Mayor probabilidad de ataque al corazón

Análisis inicial

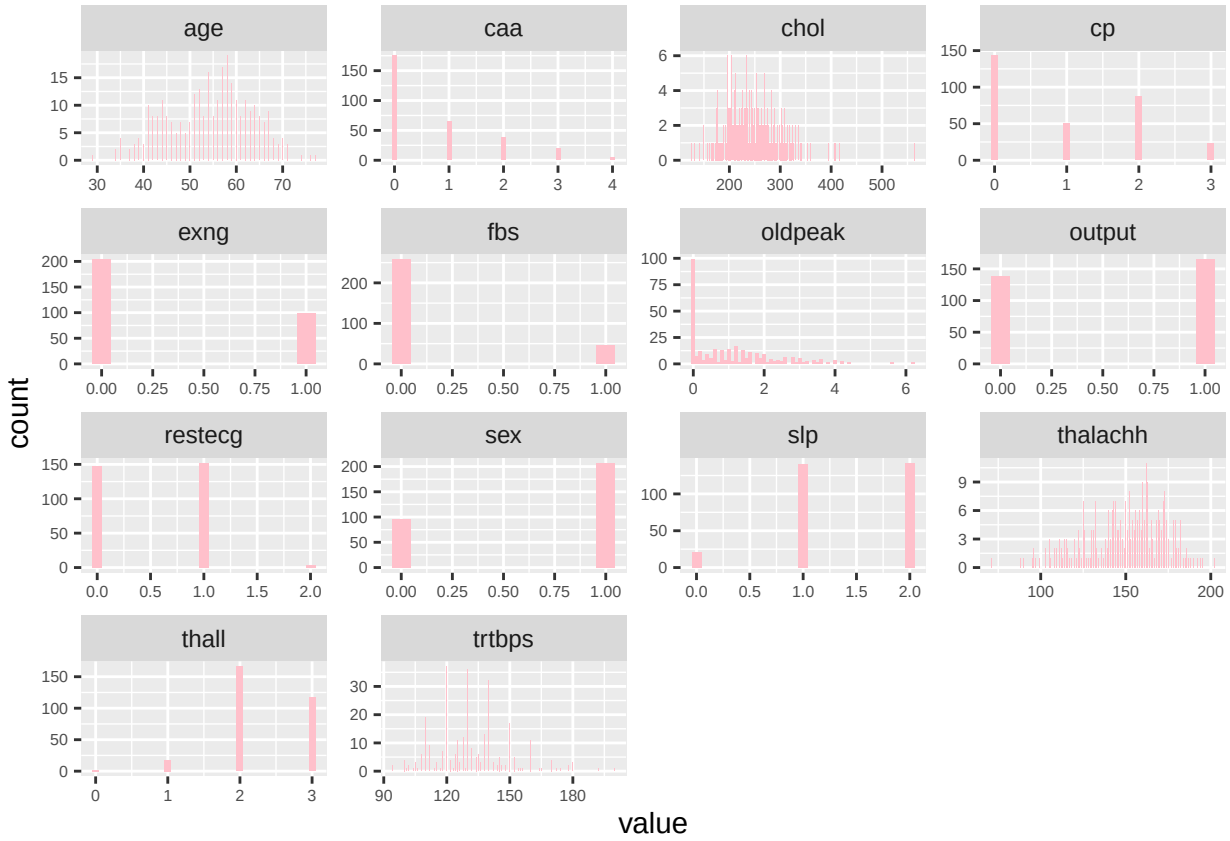
Verificamos la estructura del juego de datos principal y el tipo de datos con los que R ha interpretado cada variable, y si, corresponde a la descripción de las variables del fichero original:

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

Observamos que todas las variables se han cargado como númerica discreta a excepción de **oldpeak** que se ha cargado como númerica continua.

A continuación realizaremos una visión general del dataset con `glimpse()`.

```
## Rows: 303
## Columns: 14
## $ age      <int> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 49, 64, 58, 5~
## $ sex      <int> 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1~
## $ cp       <int> 3, 2, 1, 1, 0, 0, 1, 1, 2, 2, 0, 2, 1, 3, 3, 2, 2, 3, 0, 3, 0~
## $ trtbps   <int> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140, 130, 1~
## $ chol     <int> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239, 275, 2~
## $ fbs      <int> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0~
## $ restecg  <int> 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1~
## $ thalachh <int> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 1~
## $ exng     <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0~
## $ oldpeak  <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2, 0.2, 0~
## $ slp      <int> 0, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 0, 2, 2, 1~
## $ caa      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0~
## $ thall    <int> 1, 2, 2, 2, 2, 1, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3~
## $ output   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
```



Observamos como **sex, caa, cp, fbs, restecg, exng, slp y thall** contienen un número limitado de valores únicos, por lo que, probablemente, estén representando variables categóricas.

Comprobaremos, según la descripción oficial del dataset del punto anterior, que és cada variable y si nuestro análisis inicial es correcto.

Según la descripción oficial, observamos que nuestra suposición es correcta, y que, a excepción de **caa**, **sex, cp, fbs, restecg, exng, slp y thall** son variables categóricas. Por lo tanto, las convertiremos:

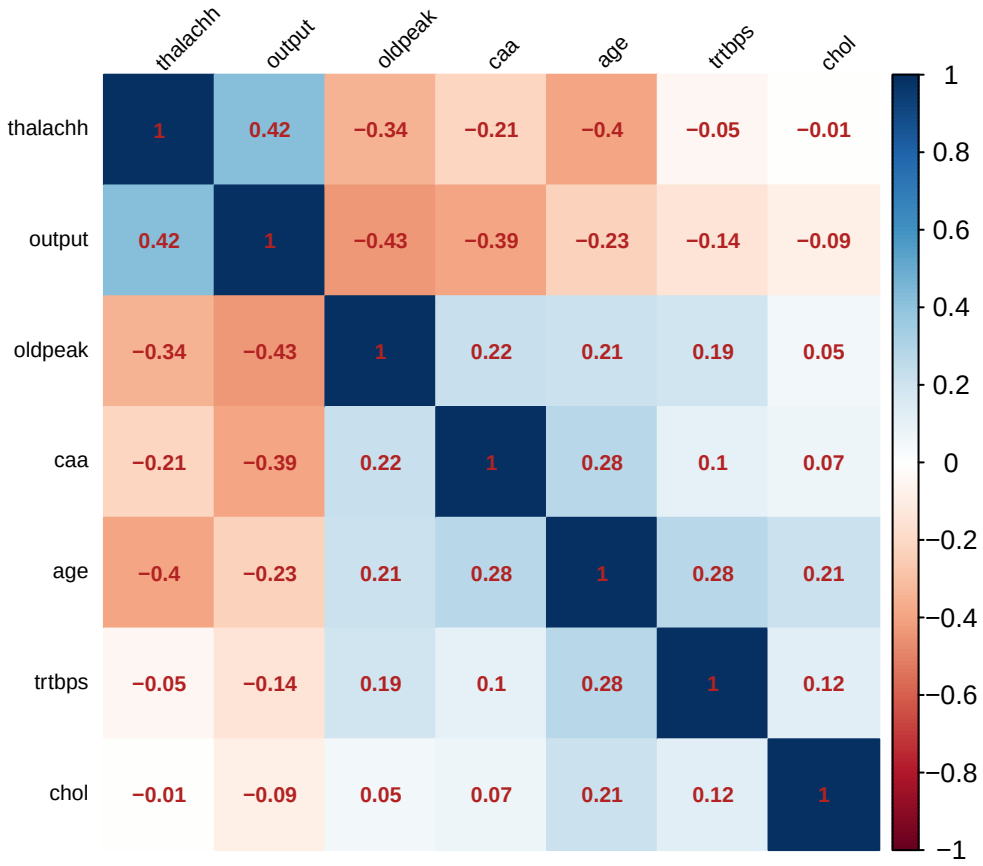
```
# Definimos las variables que hemos indentificado como categoricas
categorical_var <- c("sex", "cp", "fbs", "restecg", "exng", "slp", "thall")

# Las convertimos a factor
heartAttack <- heartAttack %>%
  mutate(across(all_of(categorical_var), factor))
```

2. Integración y selección de variables

Observando los dos ficheros csv, **`heart.csv`** tiene **14 variables** y **303 registros** mientras que **`o2Saturation.csv`** con **1 variable** y **3585 registros**.

Aunque el nivel de saturación de oxigeno pueda ser importante para los ataques cardíacos, no hay manera de juntar los dos conjuntos de datos en uno solo debido a que no hay un identificador de paciente, por lo que solo usaremos **`heart.csv`**.

Para la selección de los datos, comprobaremos la correlación entre ellas. En el caso de las variables numericas, realizaremos la correlación de Pearson:



Tanto una correlación positiva como una muy negativa son interesantes para la selección de variables. Centrándonos en la fila de la variable objetivo **`output`** se tienen los siguientes valores:

```
##      age   trtbps     chol thalachh  oldpeak      caa   output
##    -0.23    -0.14    -0.09     0.42    -0.43    -0.39     1.00
```

Se puede tomar como referencia **0.1** como umbral para comprobar las variables que no son necesarias para el estudio, siempre en valor absoluto. En este caso para el coeficiente de correlación de pearson se tienen **`age`**, **`trtbps`**, **`thalachh`**, **`oldpeak`** y **`caa`** como variables aptas y **`chol`** como poco importante.

Para las variables categóricas numéricas seria más apropiado hacer un test de **`Fisher`** o un **`Chi-squared`**.

Se va a proceder a hacer uso del test de **`Fisher`** con **`fisher.test()`**

$p = \frac{\binom{a+b}{a} \cdot \binom{c+d}{c}}{\binom{n}{a+c}}$

```r
# Creamos una lista vacía para almacenar los resultados de las pruebas de Fisher
fisher_results <- list()

# Iteramos sobre cada variable categórica
for (var in categorical_var) {
  fisher_result <- fisher.test(heartAttack[[var]], heartAttack$output)
  fisher_results[[var]] <- fisher_result
}
```

Visualizamos los resultados obtenidos del test de Fisher:

```r
print(fisher_results)
```

```
## $sex
##
##  Fisher's Exact Test for Count Data
##
## data:  heartAttack[[var]] and heartAttack$output
## p-value = 1.042e-06
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.1519598 0.4783553
## sample estimates:
## odds ratio
##  0.2731136
##
##
## $cp
##
##  Fisher's Exact Test for Count Data
##
## data:  heartAttack[[var]] and heartAttack$output
## p-value < 2.2e-16
## alternative hypothesis: two.sided
##
##
## $fbs
##
##  Fisher's Exact Test for Count Data
##
## data:  heartAttack[[var]] and heartAttack$output
## p-value = 0.6308
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.4308961 1.6975867
## sample estimates:
## odds ratio
##  0.8544825
##
##
## $restecg
##
##  Fisher's Exact Test for Count Data
```

```
##
## data:  heartAttack[[var]] and heartAttack$output
## p-value = 0.003629
## alternative hypothesis: two.sided
##
##
## $exng
##
##  Fisher's Exact Test for Count Data
##
## data:  heartAttack[[var]] and heartAttack$output
## p-value = 1.76e-14
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.07259027 0.23708719
## sample estimates:
## odds ratio
##   0.133146
##
##
## $slp
##
##  Fisher's Exact Test for Count Data
##
## data:  heartAttack[[var]] and heartAttack$output
## p-value = 1.165e-11
## alternative hypothesis: two.sided
##
##
## $thall
##
##  Fisher's Exact Test for Count Data
##
## data:  heartAttack[[var]] and heartAttack$output
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Se entiende entonces que las variables que tienen un *p-valor* por debajo de un nivel de significancia de **0.05** son consideradas buenas para ser escogidas para el análisis, es decir estas variables tienen un buen nivel estadístico de significancia y aportan información a los posibles modelos en las que se incluyan. De las variabels categoricas seleccionadas todas menos `fbs` tienen un p-valor por debajo de `0.05`.

Puesto que `fbs` no es una variable significativa se va a evitar su uso.

3. Limpieza de los datos

Volvemos a comprobar la estructura de los datos con `str()`, para verificar que los cambios realizados anteriormente se han ejecutado correctamente.

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : Factor w/ 2 levels "0","1": 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  : 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 ...
##  $ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng    : 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 ...
##  $ slp     : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
##  $ caa     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thall   : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
##  $ output  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Como se puede observar las variables categóricas ya están en tipo factor.

Y ahora se va a ver un resumen general de cada una de las variables con sus valores máximos, mínimos media, mean y cuartiles utilizando la función `summary()`. Es aquí donde en los casos numéricos se pueden ver si hay valores imposibles de cumplir tanto en máximos como en mínimos.

```
##       age        sex     cp          trtbps           chol       fbs
##  Min.   :29.00   0: 96   0:143   Min.   : 94.0   Min.   :126.0   0:258
##  1st Qu.:47.50   1:207   1: 50   1st Qu.:120.0   1st Qu.:211.0   1: 45
##  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
##  restecg    thalachh     exng       oldpeak     slp          caa
##  0:147   Min.   : 71.0   0:204   Min.   :0.00   0: 21   Min.   :0.0000
##  1:152   1st Qu.:133.5   1: 99   1st Qu.:0.00   1:140   1st Qu.:0.0000
##  2:  4   Median :153.0           Median :0.80   2:142   Median :0.0000
##          Mean   :149.6           Mean   :1.04           Mean   :0.7294
##          3rd Qu.:166.0           3rd Qu.:1.60           3rd Qu.:1.0000
##          Max.   :202.0           Max.   :6.20           Max.   :4.0000
##  thall       output
##  0:  2   Min.   :0.0000
##  1: 18   1st Qu.:0.0000
##  2:166   Median :1.0000
##  3:117   Mean   :0.5446
##          3rd Qu.:1.0000
##          Max.   :1.0000
```

De la variable `caa` se tenían identificados valores de 0 a 3, pero el valor máximos es 4.

3.1. ¿Los datos contienen ceros o elementos vacíos?

Cuando en un dataset se tienen datos nulos, hay una serie de estrategias a seguir para solucionar esto y que el juego de datos se pueda usar:

- **Eliminación de los registros**, esto a veces no es adecuado porque puede perderse mucha información que hay en otras variables que pueden ser más importantes.
- **Imputación de un valor** que puede ser: utilizar la media, la mediana, la moda, interpolación, utilización de los vecinos cercanos, u otros métodos.

Búsqueda de ceros

Tenemos algunas variables categóricas en formato numérico en nuestro conjunto de datos. Estas variables no se pueden considerar en la búsqueda de ceros, ya que el valor `0` es una de las posibles categorías para cada una de ellas. Las variables categóricas en formato numérico que ahora son factor son `sex`, `cp`, `fbs`, `restecg`, `exng`, `slp`, `thall` y la target `output`. De las cuales son dicotomicas `sex`, `fbs`, `exng` y `output`. Además existe la variabe `caa` con tres posibles valores que indican una cantidad que puede ser 0.

También en el resumen mostrado anterior se podía ver a simple vista si alguna variable tenía 0 si este fuera su valor mínimo.

Para buscar los valores nulos podemos usar `colSums()` y comprobando con un `=` como a continuación

```
selectedColumns <- c("age", "trtbps", "chol", "thalachh", "oldpeak")
colSums(heartAttack %>% select(all_of(selectedColumns)) == 0)
```

```
##      age   trtbps     chol thalachh  oldpeak
##        0        0        0        0       99
```

- **age**: Hay 0 pacientes con 0 años.
- **trtbps**: Hay 0 pacientes con 0 o sin presión arterial en reposo.
- **chol**: Hay 0 pacientes con 0 o sin medición de colesterol.
- **thalachh**: Hay 0 pacientes con 0 o sin ritmo cardíaco máximo alcanzado.
- **oldpeak**: Hay 99 pacientes con 0 o sin informar de la diferencia en segmento ST con electrocardiograma.

Búsqueda de NAs

Para buscar los valores nulos podemos usar de nuevo `colSums()` pero ahora con `is.na()`

```
colSums(is.na(heartAttack))
```

```
##      age      sex       cp   trtbps     chol      fbs  restecg thalachh
##        0        0        0        0        0        0        0        0
##     exng  oldpeak      slp      caa    thall   output
##        0        0        0        0        0        0
```

Como se puede observar **no hay valores `NA`** en este dataset, otra comprobación sería buscar valores en blanco, pero esto se haría si hubiera variables categoricas que fueran cadenas, en este caso no es necesario ya que no hay ningún valor como texto.

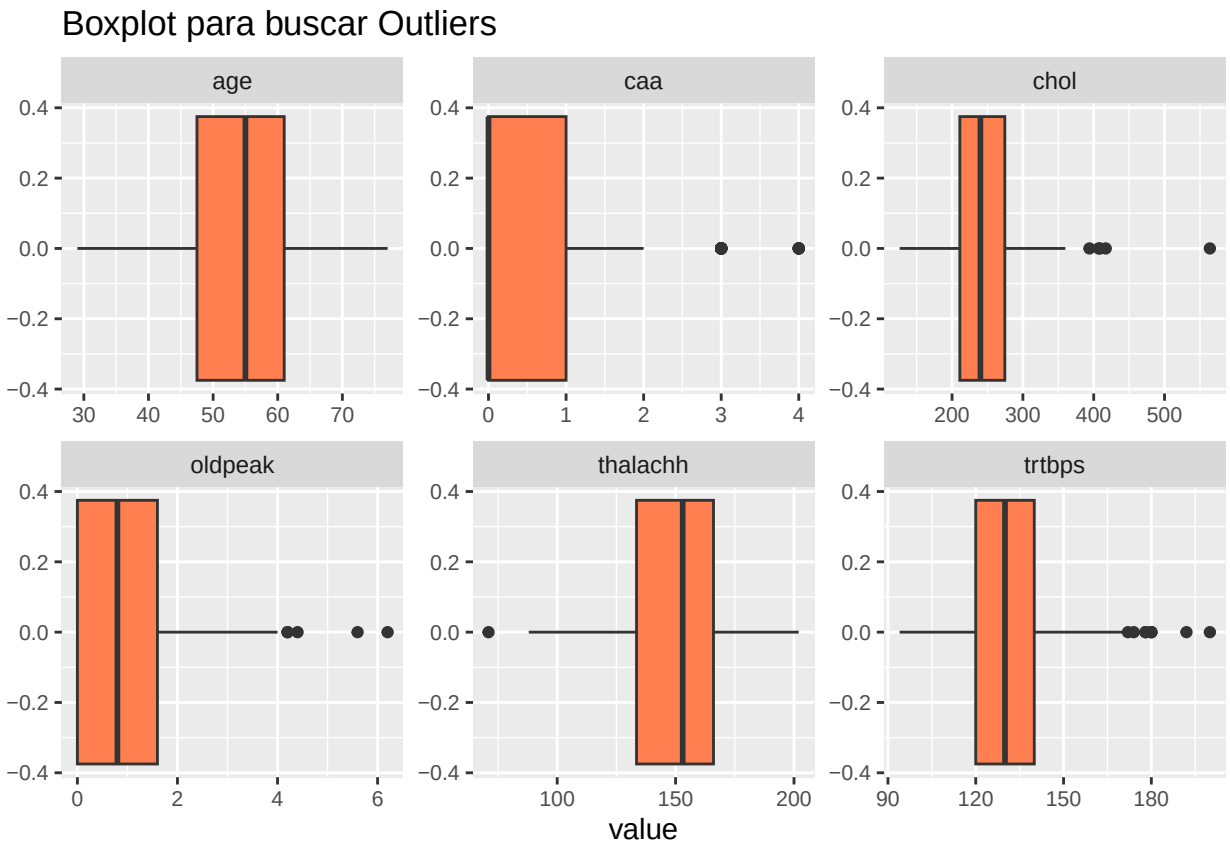
3.2. Identifica y gestiona los valores extremos

Primero en las variables numéricas vamos a comprobar cuantos valores atípicos hay de cada una

```
##      age      caa     chol  oldpeak thalachh   trtbps
##        0       25        5        5        1        9
```

Como se puede observar en las variables numéricas, `age` no tiene valores atípicos.

Ahora se visualizarán los valores atípicos de las variables numéricas.



Los valores extremos son:

- Para `caa`: 3, 4.
- Para `chol`: 394, 407, 409, 417, 564.
- Para `oldpeak`: 4.2, 4.4, 5.6, 6.2.
- Para `thalachh`: 71.
- Para `trtbps`: 172, 174, 178, 180, 192, 200.

FIXME

4. Análisis de los datos

4.1. Selección de los grupos de datos que se quieren analizar/comparar

Deseamos conocer la relación que existe entre las siguientes variables:

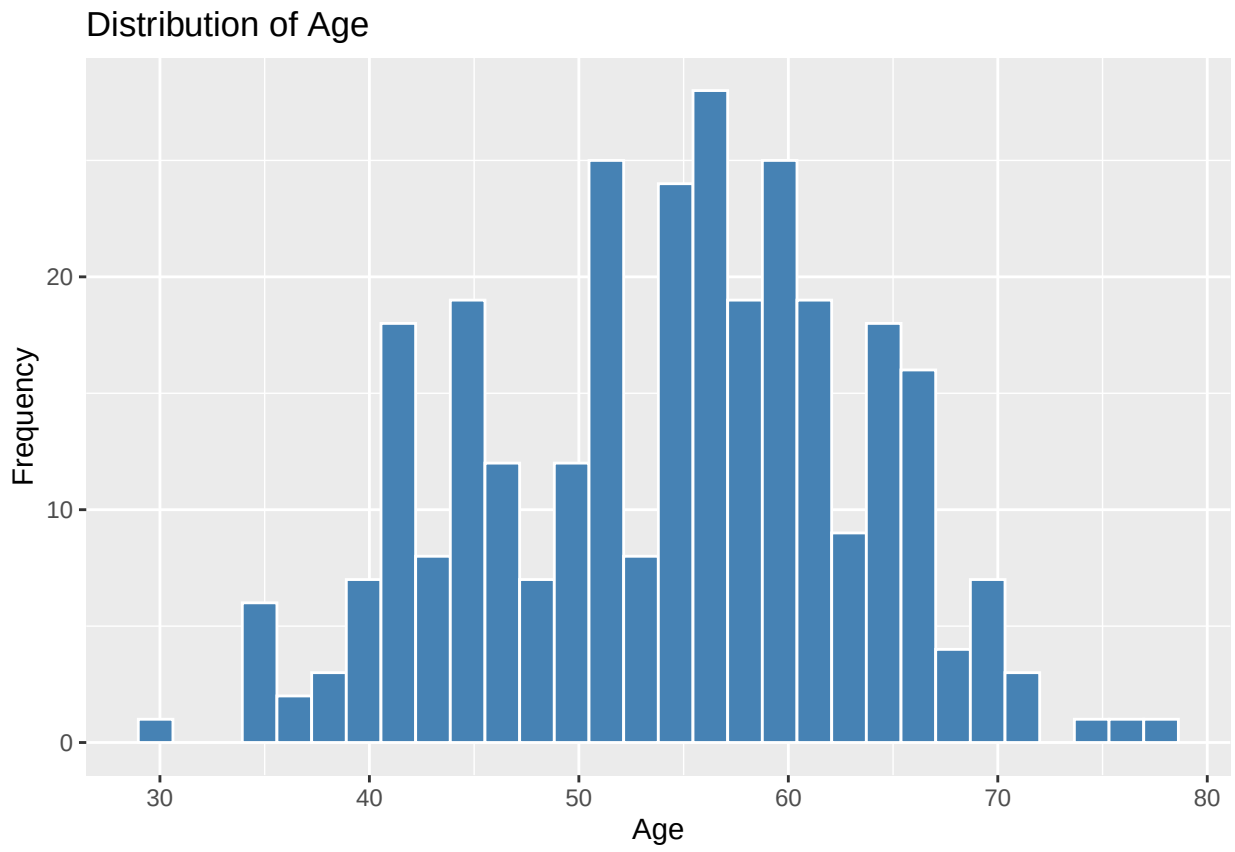
- **sex**, **cp**: Queremos conocer si existen diferencias significativas entre el tipo de dolor en el pecho que experimentan las observaciones con infarto en función del sexo.
- **age**:Queremos conocer si la media de las variables númericas **trtbps, chol** son las mismas para los grupos de datos **age** tras realizar una discretización de esta variable.
- **output vs variables**: Queremos aproximar la relación de dependencia que existe entre la probabilidad de sufrir un infarto y las variables del dataset.

4.2. Comprobación de la normalidad y homogeneidad de la varianza

sex, cp y output: No es necesario comprobar la normalidad y homogeneidad de la varianza, ya que aplicaremos el test chi-cuadrado, que se trata de un test no parametrico.

age: Seleccionaremos el test a aplicar en función de la normalidad y la homogeneidad de la varianza de los grupos a comparar.

El primer paso es discretizar la variable age:



Discretizamos la variable en tres grupos: `Jovenes`, `Media` y `Vejez`

```r
# Calculamos los cuantiles
age_breaks <- c(0, 45, 60, 80)

# Discretizamos la variable age en grupos
heartAttack$age_discretized <-
  cut(
    heartAttack$age,
    breaks = age_breaks,
    labels = c("Jovenes", "Media", "Vejez"),
    include.lowest = TRUE
  )
```

Visualizamos el mínimo y máximo valor para age de cada grupo.

```r
# Obtener el máximo y mínimo de la variable "age_discretized" en el conjunto de datos "heartAttack"
max_min_age <- tapply(heartAttack$age, heartAttack$age_discretized, range)

# Obtener el máximo y mínimo para el grupo "tercio1"
max_jovenes <- max_min_age[["Jovenes"]][2]
min_jovenes <- max_min_age[["Jovenes"]][1]

# Obtener el máximo y mínimo para el grupo "tercio2"
max_medio <- max_min_age[["Media"]][2]
min_medio <- max_min_age[["Media"]][1]

# Obtener el máximo y mínimo para el grupo "tercio3"
max_vejez <- max_min_age[["Vejez"]][2]
min_vejez <- max_min_age[["Vejez"]][1]

cat("Jovenes -", "Mínimo:", min_jovenes,"Máximo:", max_jovenes, "\n")
```

```
## Jovenes - Mínimo: 29 Máximo: 45
```

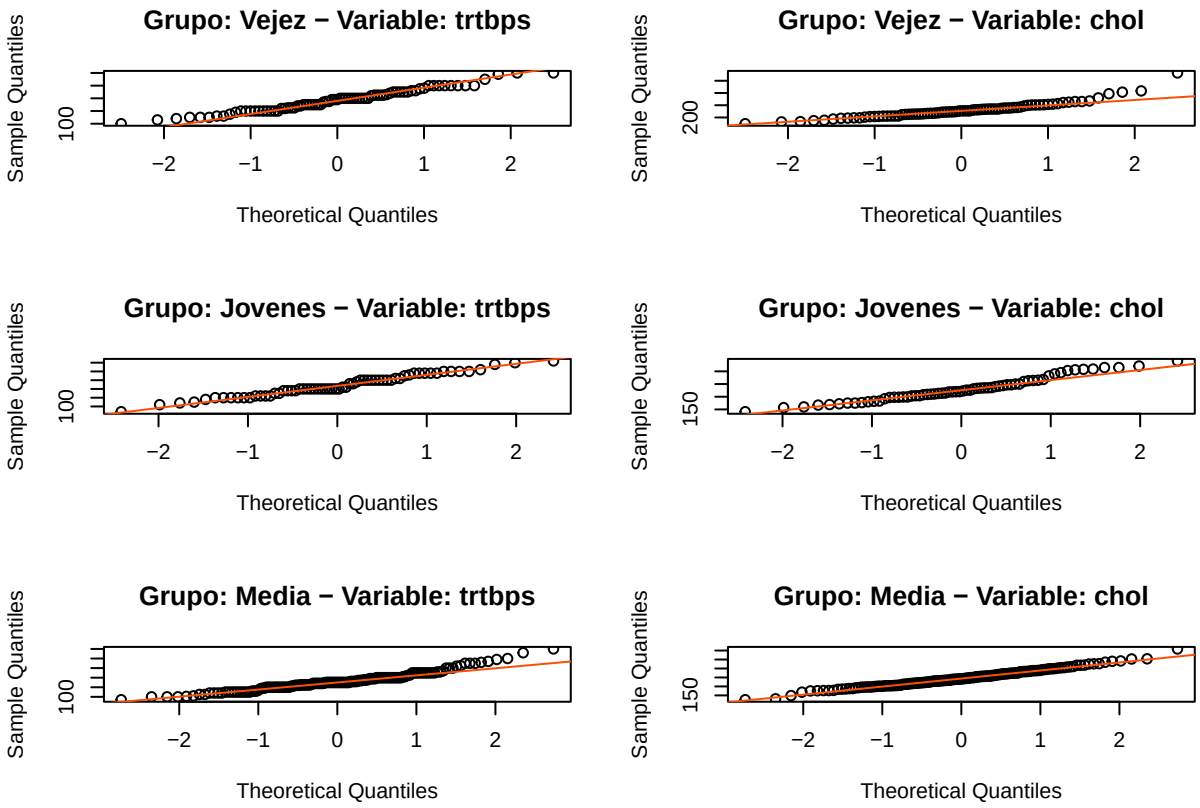
```r
cat("Medio -", "Mínimo:", min_medio,"Máximo:", max_medio, "\n")
```

```
## Medio - Mínimo: 46 Máximo: 60
```

```r
cat("Vejez -", "Mínimo:", min_vejez,"Máximo:", max_vejez, "\n")
```

```
## Vejez - Mínimo: 61 Máximo: 77
```

Ahora comprobamos si las variables **trtbps, chol** presentan una distribución normal de manera visual, a través del gráfico Q-Q:



De manera visual, parece que la variable **colesterol** en el grupo Vejez, se aleja de la recta normal en la parte derecha el gráfico. Lo mismo se observa para la variable **trtbps** en el grupo de edad Media.

Ahora realizaremos una evaluación más cuantitativa de la distribución, mediante el test Shapiro-Wilk:

```r
# Comprobamos la normalidad por grupo
group_names <- unique(heartAttack$age_discretized)

for (group in group_names) {
  group_data <- subset(heartAttack, age_discretized == group)

  shapiro_test_chol <- shapiro.test(group_data$chol)
  shapiro_test_trtbps <- shapiro.test(group_data$trtbps)

  cat("Grupo:", group, "\n")
  cat("El p-value de chol es:", shapiro_test_chol$p.value, "\n")
  cat("El p-value de trtbps es:", shapiro_test_trtbps$p.value, "\n")

  if (shapiro_test_chol$p.value >= 0.05) {
    cat("La variable 'colesterol' en el grupo", group, "sigue una distribución normal.\n")
  } else {
    cat("La variable 'colesterol' en el grupo", group, "no sigue una distribución normal.\n")
  }

   if (shapiro_test_trtbps$p.value >= 0.05) {
    cat("La variable 'trtbps' en el grupo", group, "sigue una distribución normal.\n")
  } else {
```

```
    cat("La variable 'trtbps' en el grupo", group, "no sigue una distribución normal.\n")
  }
    cat("\n")
}
```

```
## Grupo: Vejez
## El p-value de chol es: 5.001049e-06
## El p-value de trtbps es: 0.1739525
## La variable 'colesterol' en el grupo Vejez no sigue una distribución normal.
## La variable 'trtbps' en el grupo Vejez sigue una distribución normal.
##
## Grupo: Jovenes
## El p-value de chol es: 0.07891974
## El p-value de trtbps es: 0.2936327
## La variable 'colesterol' en el grupo Jovenes sigue una distribución normal.
## La variable 'trtbps' en el grupo Jovenes sigue una distribución normal.
##
## Grupo: Media
## El p-value de chol es: 0.2900424
## El p-value de trtbps es: 2.96224e-05
## La variable 'colesterol' en el grupo Media sigue una distribución normal.
## La variable 'trtbps' en el grupo Media no sigue una distribución normal.
```

Observamos como se cumple lo que hemos visualizado en los gráficos Q-Q y para los grupos vejez y media, la variable chol y trtbps no presentan una distribución normal. Por lo que, descartamos el test de ANOVA de una via y aplicaremos Kruskal-Wallis que no asume una distribución normal en los datos.

No es necesario comprobar la homogeneidad de los datos ya que al no cumplir el criterio de normalidad, aplicaremos un test no parámetrico.

OUTPUT VS VARIABLES: Aproximaremos la relación de dependencia entre la variable dependiente output y el resto de variables. En este caso, no es necesario que las variables presenten una distribución normal, ya que emplearemos la regresión logística, y estos supuestos no son requisitos para el modelo.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos

CHI-CUADRADO

Para analizar las diferencias entre el tipo de dolor en el pecho y el sexo en las observaciones con infarto realizaremos la prueba de chi-cuadrado.

El primer paso es seleccionar únicamente las observaciones con output 1.

```
infarto <- subset(heartAttack, output== 1)
```

Calculamos, la frecuencia, en una tabla, con el tipo de dolor en el pecho y el sexo.

```
tabla <- table(infarto$cp, infarto$sex)
tabla
```

```
##
##      0  1
##   0 18 21
```

```
##   1 16 25
##   2 34 35
##   3  4 12
```

Realizamos la prueba chi-cuadrado:

```
# Prueba de chi-cuadrado
chi_square <- chisq.test(tabla)

# Imprimimos los resultados
print(chi_square)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tabla
## X-squared = 3.6066, df = 3, p-value = 0.3072
```

El resultado de p-value superior a 0,05 indica que no se encuentran diferencias significativas para el tipo de dolor en el pecho y el sexo dentro de la población que sufre un infarto.

KRUSKAL-WALLIS

Para comparar las variables **trtbps** y **chol** entre los grupos de edad **Jovenes, Media y Vejez**, y como sabemos que no se cumple en supuesto de normalidad de los datos en cada grupo, aplicaremos Kruskal_Wallis.

Nuestra hipotesis son:

** Hipotesis nula (HO): No hay diferencias significativas de la media del valor trtbps y chol entre los diferentes grupos de edad.

** Hipotesis alternativa (H1): Hay diferencias significativas de la media del valor trtbps y chol entre los diferentes grupos de edad.

Realizamos el test de Kruskal-Wallis para la presión arterial y los diferentes grupos de edad:

```
kruskal.test(trtbps~age_discretized, data=heartAttack)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  trtbps by age_discretized
## Kruskal-Wallis chi-squared = 20.185, df = 2, p-value = 4.138e-05
```

Realizamos el test de Kruskal-Wallis para el colesterol y los diferentes grupos de edad:

```
kruskal.test(chol~age_discretized, data=heartAttack)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  chol by age_discretized
## Kruskal-Wallis chi-squared = 10.993, df = 2, p-value = 0.004101
```

Los valores de p obtenidos en ambos test, son inferiores a 0.05, por lo que podemos rechazar la hipotesis nula y concluir que el valor de colesterol y presión sanguinea varia en función del grupo de edad de las observaciones.

REGRESIÓN LOGISTICA

Por último, aproximaremos la relación de dependencia entre la variable dependiente **output** y **el resto de variables** mediante una regresión logistica.

Dividimos los datos entre train y test.

```
#Utilizaremos el 70% para train y el 30% para test
set.seed(23)
train_index <- sample(1:nrow(heartAttack), nrow(heartAttack) * 0.7)
train <- heartAttack[train_index, ]
test <- heartAttack[-train_index, ]
```

Ajustamos el modelo de regresión logistica.

```
model <- glm(output ~ ., data = train, family = binomial)
```

Visualizamos el resultado del modelo:

```
# Obtenemos los resultados del modelo
summary(model)
```

```
##
## Call:
## glm(formula = output ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4284  -0.4601   0.1278   0.4834   2.4724
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.013246   4.161755  -0.003 0.997461
## age                   0.066110   0.058969   1.121 0.262244
## sex1                 -1.620719   0.623462  -2.600 0.009335 **
## cp1                   1.626966   0.698908   2.328 0.019919 *
## cp2                   1.944191   0.568521   3.420 0.000627 ***
## cp3                   2.107814   0.764826   2.756 0.005852 **
## trtbps               -0.019821   0.013263  -1.494 0.135056
## chol                 -0.003682   0.004510  -0.816 0.414244
## fbs1                  0.379609   0.691762   0.549 0.583172
## restecg1              0.676121   0.463103   1.460 0.144296
## restecg2             -0.780613   2.746399  -0.284 0.776233
## thalachh              0.010885   0.014418   0.755 0.450296
## exng1                -0.836982   0.518187  -1.615 0.106265
## oldpeak              -0.466953   0.265264  -1.760 0.078352 .
## slp1                 -0.765474   1.066103  -0.718 0.472751
## slp2                 -0.325458   1.178365  -0.276 0.782399
## caa                  -0.604750   0.218450  -2.768 0.005634 **
## thall1                0.930690   2.301646   0.404 0.685949
```

```
## thall2                1.621853   2.199449   0.737 0.460885
## thall3                0.173363   2.207966   0.079 0.937417
## age_discretizedMedia -1.114205   0.933980  -1.193 0.232883
## age_discretizedVejez -1.791554   1.578239  -1.135 0.256308
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 293.42  on 211  degrees of freedom
## Residual deviance: 145.52  on 190  degrees of freedom
## AIC: 189.52
##
## Number of Fisher Scoring iterations: 6
```

Observamos que las variables que son estadísticamente significativas para output son **cp, restecg, oldpeak y caa**.

Por lo tanto, volvemos a realizar un modelo con solo esas variables.

```
# Ajustamos un nuevo modelo con las variables significativas
model_2 <- glm(output ~ cp + restecg + oldpeak + caa, data = train, family = binomial)

summary(model_2)
```

```
##
## Call:
## glm(formula = output ~ cp + restecg + oldpeak + caa, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4296  -0.6862   0.3276   0.6732   2.2633
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.02819    0.36506   0.077 0.938458
## cp1          2.04886    0.54342   3.770 0.000163 ***
## cp2          2.30319    0.44975   5.121 3.04e-07 ***
## cp3          2.44321    0.68860   3.548 0.000388 ***
## restecg1     0.56653    0.36362   1.558 0.119225
## restecg2     0.17278    1.87059   0.092 0.926405
## oldpeak     -0.83292    0.20339  -4.095 4.22e-05 ***
## caa         -0.70488    0.18123  -3.889 0.000100 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 293.42  on 211  degrees of freedom
## Residual deviance: 190.62  on 204  degrees of freedom
## AIC: 206.62
##
## Number of Fisher Scoring iterations: 5
```

Evaluamos el modelo. Predeciremos la variable output para el conjunto de test, y compararemos con los resultados reales. Visualizaremos la matriz de confusión.

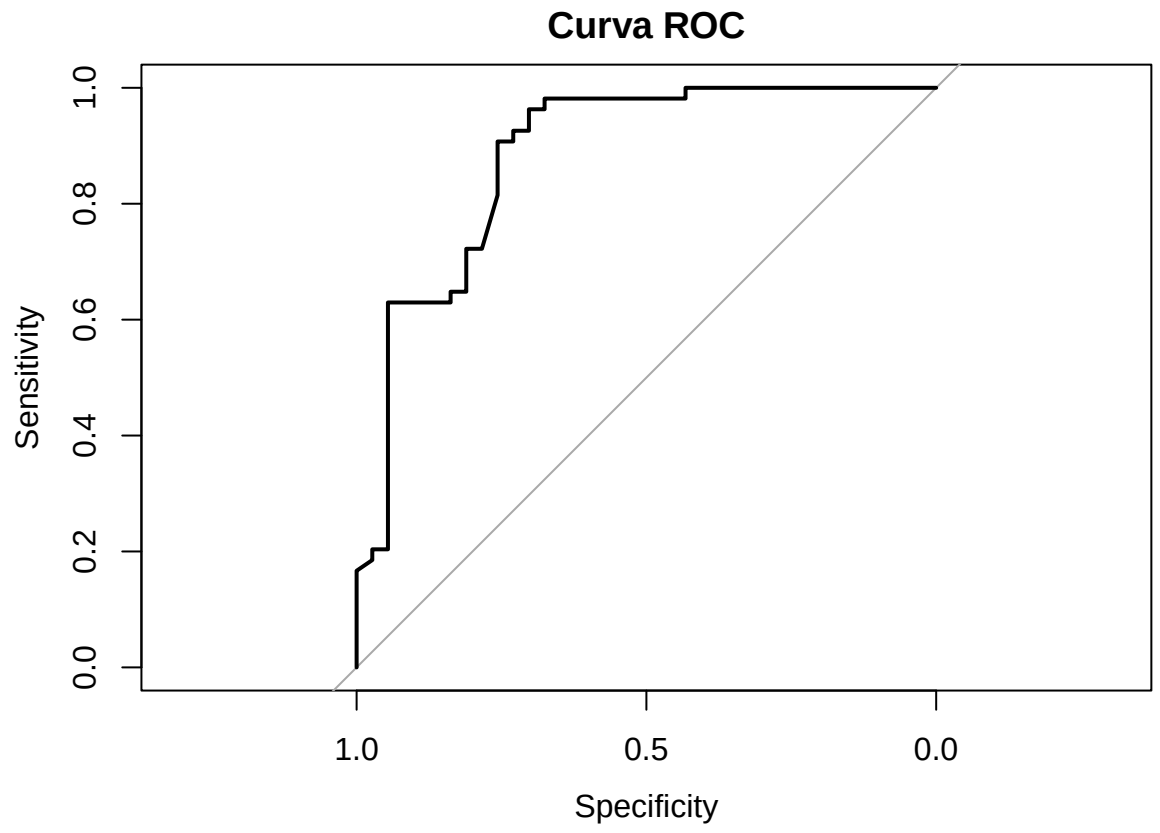
```
#Predecimos
predicted <- predict(model_2, newdata = test, type = "response") > 0.5

#realizamos la matriz de confusion
confusion_matrix <- table(predicted, test$output)


# Normalizamos la matriz de confusión
normalized_confusion_matrix <- prop.table(confusion_matrix, margin = 1)

# Visualizamos la matriz de confusión normalizada
print(normalized_confusion_matrix)
```

```
##
## predicted         0         1
##     FALSE 0.8235294 0.1764706
##     TRUE  0.1578947 0.8421053
```

Por ultimo, calcuaremos la curva ROC y evaluamos el modelo con el coeficiente AUC, que indica el valor del area bajo la curva.

Visualizamos la curva:

```
plot(roc_obj, main = "Curva ROC")
```



Visualizamos el valor del coeficiente bajo la curva:

```
auc
```

```
##
## Call:
## roc.default(response = test$output, predictor = predicted_probs)
##
## Data: predicted_probs in 37 controls (test$output 0) < 54 cases (test$output 1).
## Area under the curve: 0.8829
```

Del resultado del modelo podemos concluir que:

La ecuación del modelo ajustado es logit(p) = 0.4046 + 1.5603 * cp1 + 2.1369 * cp2 + 1.8318 * cp3 + 0.6486 * restecg1 - 13.0644 * restecg2 - 0.9669 * oldpeak - 0.6658 * caa

- Las variables que presentan significancia respecto la variable de salida (infarto) son cp, restecg, oldpeak y caa
- Los coeficientes de la ecuación del modelo indican por cada unidad que aumenta es vararíable, cuanto aumenta el log-odds de la variable output.
- La matriz de confusión normalizada indica que el modelo predice el 84,21% de los valores falsos correctamente, con una tasa de error del 15,79%. Igualmente el modelo predice correctamente el 77,36% de los casos de infarto, con una tasa de error del 22,64%.
- El valor de AUC de 0,8769 indica un buen rendimiento del modelo.

5. Representación de los resultados

FIXME

6. Resolución del problema

FIXME

7. Código

FIXME

8. Vídeo

FIXME