Limpieza y Análisis de Datos

Lucas Gómez Torres y Joan Amengual Mesquida

13 de enero, 2023

Índice General

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

Actualmente cada vez sufren más personas ataques al corazón originados por diferentes factores como pueden ser el exceso de colesterol, el nivel de azúcar en la sangre, el consumo de tabaco, la presión arterial, la obesidad, la edad o la falta de ejercicio, entre muchos otros más, que pueden dar lugar a un daño permanente en el corazón como la insuficiencia cardiaca o a la muerte.

Por ello, los ataques al corazón son un problema muy grave que hay que intentar prevenir, analizando las diferentes variables que pueden influir a la hora de que una persona sufra un ataque al corazón o no, pudiendo responder a preguntas como por ejemplo:

- ¿Los hombres son más probables a sufrir un ataque?
- ¿ El nivel de azúcar en sangre es determinante para que una persona pueda padecer un ataque ?
- ¿ Las personas mayores tienen más probabilidad de sufrir un ataque?
- ¿ Qué factor es el más influye en un ataque ?

El conjunto de datos está dividido en dos subconjuntos de datos:

- *heart.csv*: contiene toda la información sobre los pacientes, incluyendo si finalmente sufrieron un ataque al corazón o no. Tiene 303 observaciones y 14 atributos. De estos 14 atributos, 13 son variables independientes y 1 la variable dependiente (nuestra variable objetivo que servirá para construir un modelo de aprendizaje supervisado que nos permita predecir si un paciente tendrá un ataque al corazón o no). A continuación, se describen todos los atributos de este dataset:
  - **age**: Variable de tipo numérica. Determina la edad de la persona.
  - **sex**: Variable de tipo numérica. Refleja el género de la persona *(1 = masculino, 0 = femenino)*.
  - **cp**: Variable de tipo numérica. Identifica el tipo de dolor en el pecho *(0 = angina típica, 1 = angina atípica, 2 = dolor no anginoso, 3 = asintomático)*.
  - **trtbps**: Variable de tipo numérica. Indica la presión arterial en reposo en mg/dl.
  - **chol**: Variable de tipo numérica. Hace referencia al nivel de colesterol en mg/dl.
  - **fbs**: Variable de tipo numérica. Indica si el nivel de azúcar en sangre en ayunas es mayor a 120 mg/dl *(1 = verdadero, 0 = falso)*.
  - **restecg**: Variable de tipo numérica. Muestra los resultados electrocardiográficos en reposo *(0 = normal, 1 = anomalía de onda ST-T (inversiones de onda T y/o elevación o depresión ST de > 0,05 mV), 2 = hipertrofia ventricular izquierda probable o definida por los criterios de Estes)*.
  - **thalachh**: Variable de tipo numérica. Determina la frecuencia cardiaca máxima alcanzada.
  - **exng:**: Variable de tipo numérica. Indica si la angina ha sido inducida por el ejercicio *(1 = sí, 0 = no)*.
  - **oldpeak**: Variable de tipo numérica. Señala la depresión ST inducida por el ejercicio en relación con el descanso.
  - **slp**: Variable de tipo numérica. Muestra la pendiente del segmento ST de ejercicio máximo *(0 = inclinación hacia abajo, 1 = plano, 2 = inclinación hacia arriba)*.
  - **caa**: Variable de tipo numérica. Indica el número de buques principales *(0, 1, 2, 3)*.

  - **thall**: Variable de tipo numérica. Señala el ratio de un trastorno sanguíneo llamado talasemia *(0 = no tiene, 1 = defecto fijo (sin flujo sanguíneo en alguna parte del corazón), 2 = flujo sanguíneo normal, 3 = defecto reversible (se observa un flujo sanguíneo, pero no es normal)).*
  - **output**: Variable de tipo numérica. Indica si el paciente sufre un ataque al corazón o no (0 = No, 1 = Sí). Se trata de la variable objetivo o dependiente que pretenderemos predecir.

- *o2Saturation.csv*: contiene 3585 observaciones sobre los niveles de oxígeno en la sangre de distintos pacientes y solo tiene 1 atributo.

2 Integración y selección de los datos de interés a analizar.

Puede ser el resultado de adicionar diferentes datasets o una subselección útil de los datos originales, en base al objetivo que se quiera conseguir.

En este apartado se van a cargar ambos conjuntos de datos, para decidir si se van a unificar ambos o no, o si nos vamos a centrar en unos pasajeros concretos limitando el número de registros o de características con el fin de reducir el dataset. Además, en el dataset de *heart.csv* se van a renombrar los atributos para que se entiendan mejor y sean más intuitivos a la hora de utilizarlos más adelante.

```
# Se carga el dataset
heart_data <- read.csv("heart.csv", header = TRUE)

# Modificamos los nombres de las variables para que sean más intuitivos
colnames(heart_data) <- c("age","sex","chest_pain_type","resting_blood_pressure",
                          "cholesterol",    "fasting_blood_sugar","rest_ecg_type",
                          "max_heart_rate_achieved","exercise_induced_angina",
                          "st_depression", "st_slope_type", "num_major_vessels",
                       "thalassemia_type","heart_attack")

# Dimensión del dataset
dim(heart_data)
```

```
## [1] 303  14
```

```
# Se carga el dataset
O2_saturation <- read.csv("o2Saturation.csv", header = TRUE)

# Dimensión del dataset
dim(O2_saturation)
```

```
## [1] 3585    1
```

Podemos observar que ambos conjuntos de datos tienen dimensiones diferentes. El que contiene los niveles de oxígeno en la sangre consta de 3.585 observaciones, es decir, diferentes niveles de oxígeno para 3.585 pacientes, en cambio, el otro, contiene información sobre 303 pacientes y 14 características distintas. Como ya tenemos suficientes características en el dataset de *heart.csv* con las que poder realizar un estudio detallado y completo a las preguntas que hemos planteado al principio, se va a optar por descartar el otro conjunto y perder este atributo adicional de los pacientes.

En el caso de haber querido unificarlos y por lo tanto añadir otro atributo al dataset de *heart.csv* (saturación de oxígeno), se podría haber utilizado la función *merge* permitiéndonos fusionarlos de forma horizontal. Posteriormente, se podría comprobar que no existen inconsistencias ni duplicidades en los registros con la función *duplicated* o *unique*. No obstante, no existe un identificador único para cada uno de los pacientes como podría ser un id o un nombre, por lo que suponemos que podría haber dos pacientes con los mismos valores de atributos. Asimismo comprobaremos si hay muchos registros duplicados con el fin de que no pueda afectar significamente en los análisis posteriores.

```
# Comprobamos si existen registros duplicados con los mismos valores en todos los campos
# (dado que no tenemos identificador) Y contamos cuántos son

nrow(heart_data[duplicated(heart_data), ])
```

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

```
# Vemos los registros que están duplicados
heart_data[duplicated(heart_data), ]
```

```
##     age sex chest_pain_type resting_blood_pressure cholesterol
## 165  38   1               2                    138         175
##     fasting_blood_sugar rest_ecg_type max_heart_rate_achieved
## 165                   0             1                     173
##     exercise_induced_angina st_depression st_slope_type num_major_vessels
## 165                       0             0             2                 4
##     thalassemia_type heart_attack
## 165                2            1
```

Dado que solo existe un registro duplicado, con los mismos valores en todos los campos, no se va a eliminar porque es un porcentaje muy bajo del total y no afectará de manera significativa a los resultados que obtendremos más adelante. Además, al ser solo un registro, podría ser el caso de que esos dos pacientes fueran distintos y tuvieran las mismas características. Si tuviéramos muchos más, entonces seguramente serían los mismos pacientes y tendríamos que eliminarlos.

A continuación, se muestran algunos registros e información general de los datos que servirá para posteriomente proceder a la limpieza y conversión de los datos.

```
# Mostramos los tipos de datos de las variables tal y como las interpreta R
sapply(heart_data,class)
```

```
##                      age                      sex         chest_pain_type
##                "integer"                "integer"                "integer"
##  resting_blood_pressure              cholesterol     fasting_blood_sugar
##                "integer"                "integer"                "integer"
##           rest_ecg_type max_heart_rate_achieved exercise_induced_angina
##                "integer"                "integer"                "integer"
##           st_depression           st_slope_type       num_major_vessels
##                "numeric"                "integer"                "integer"
##        thalassemia_type             heart_attack
##                "integer"                "integer"
```

```
# Mostramos un resumen de los datos
summary(heart_data)
```

```
##       age             sex         chest_pain_type resting_blood_pressure
##  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
##   cholesterol    fasting_blood_sugar rest_ecg_type    max_heart_rate_achieved
##  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
##  exercise_induced_angina st_depression  st_slope_type   num_major_vessels
##  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
##  thalassemia_type  heart_attack
##  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
```

```r
# Se muestran las 4 primeras observaciones de los datos
head(heart_data,4)
```

```
##   age sex chest_pain_type resting_blood_pressure cholesterol
## 1  63   1               3                   145         233
## 2  37   1               2                   130         250
## 3  41   0               1                   130         204
## 4  56   1               1                   120         236
##   fasting_blood_sugar rest_ecg_type max_heart_rate_achieved
## 1                   1            0                     150
## 2                   0            1                     187
## 3                   0            0                     172
## 4                   0            1                     178
##   exercise_induced_angina st_depression st_slope_type num_major_vessels
## 1                       0          2.3             0                 0
## 2                       0          3.5             0                 0
## 3                       0          1.4             2                 0
## 4                       0          0.8             2                 0
##   thalassemia_type heart_attack
## 1                1            1
## 2                2            1
## 3                2            1
## 4                2            1
```

Por último, para nuestro análisis no se van a descartar registros porque no nos vamos a centrar en un tramo de edad concreto, sexo o una cantidad de colesterol, sino que se van a considerar a todos los pacientes con todas sus características para extraer el mayor número de conclusiones posibles teniendo en cuenta todos los atributos.

3 Limpieza de los datos.

3.1 ¿Los datos contienen ceros o elementos vacíos? Gestiona cada uno de estos casos.

3.1.1 Caso: Ceros

3.1.2 Caso: Elementos Vacíos

IDEA: Para imputar valores perdidos se podría utilizar KNN.

```r
#colMeans(is.na(heart_data))

#colMeans(heart_data == "")
```

3.1.3 Conversión y adaptación de los datos

Se van a realizar algunas conversiones de los tipos de algunas variables para realizar un análisis más eficiente y que nos facilite la interpretación de los resultados.

Primero convertiremos las siguientes variables numéricas a categóricas:

```r
# Transformamos a tipo factor las siguientes variables
heart_data$sex <- factor(heart_data$sex, levels = c(0,1), labels=
                           c("Femenino", "Masculino"))
heart_data$chest_pain_type <- factor(heart_data$chest_pain_type, levels = c(0,1,2,3), labels=
                                       c("Angina típica", "Angina atípica",
                                        "Dolor no anginoso","Asintomático"))

heart_data$fasting_blood_sugar <- factor(heart_data$fasting_blood_sugar, levels = c(0,1),
                                         labels=
                                           c("Azúcar Bajo", "Azúcar Alto"))

heart_data$rest_ecg_type <- factor(heart_data$rest_ecg_type, levels = c(0,1,2), labels=
                                     c("Normal", "Anomalía de onda ST-T",
                                      "Hipertrofia ventricular izquierda"))

heart_data$exercise_induced_angina <- factor(heart_data$exercise_induced_angina,
                                             levels = c(0,1), labels= c("No", "Sí"))

heart_data$st_slope_type <- factor(heart_data$st_slope_type, levels = c(0,1,2),
                                   labels= c("Baja", "Normal","Alta"))
heart_data$thalassemia_type <- factor(heart_data$thalassemia_type, levels = c(0,1,2,3),
                                      labels= c("Inexistente", "Fijo",
                                          "Normal","Reversible"))
```

También se pueden aplicar otro tipo de conversiones como por ejemplo la normalización *z-score* que resta la media a la variable y la divide por su desviación estándar. Usaremos esta normalización usando la función *scale* para normalizar las variables cuantitativas.

```r
# Indices de las variables cuantitativas
idx_var_cuant <- c(1,4,5,8,10,12)

# Normalización variables cuantitativas
heart_norm <- scale(heart_data[,idx_var_cuant])
```

Es posible que se tengan que utilizar más adelanete será estos datos normalizados, sin embargo, se van a mantener sin normalizar ya que para mostrar los resultados resulta más intuitivo verlos en su escala natural.

En el caso de las variables que no presenten una distribución normal, una opción sería realizar transformaciones de tipo Box-Cox para poder mejorar su normalidad y su homocedasticidad.

Asimismo, para algunas variables como por ejemplo la edad del paciente, sería interesante realizar un proceso de discretización. Esto nos permitiría agrupar las edades en diferentes grupos y poder sacar conclusiones que nos aporten un valor simbólico más allá de solo un número, aportándonos mayor información. HACERRRRR JEJEJ

3.2 Identifica y gestiona los valores extremos.

4 Análisis de los datos.

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

4.2 Comprobación de la normalidad y homogeneidad de la varianza.

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

5 Representación de los resultados a partir de tablas y gráficas.

Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica.

6 Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

7 Código.

Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

8 Vídeo.

Realizar un breve vídeo explicativo de la práctica (máximo 10 minutos), donde ambos integrantes del equipo expliquen con sus propias palabras el desarrollo de la práctica, basándose en las preguntas del enunciado para justificar y explicar el código desarrollado. Este vídeo se deberá entregar a través de un enlace al Google Drive de la UOC, junto con enlace al repositorio Git entregafo.