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
```

```
# 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
```

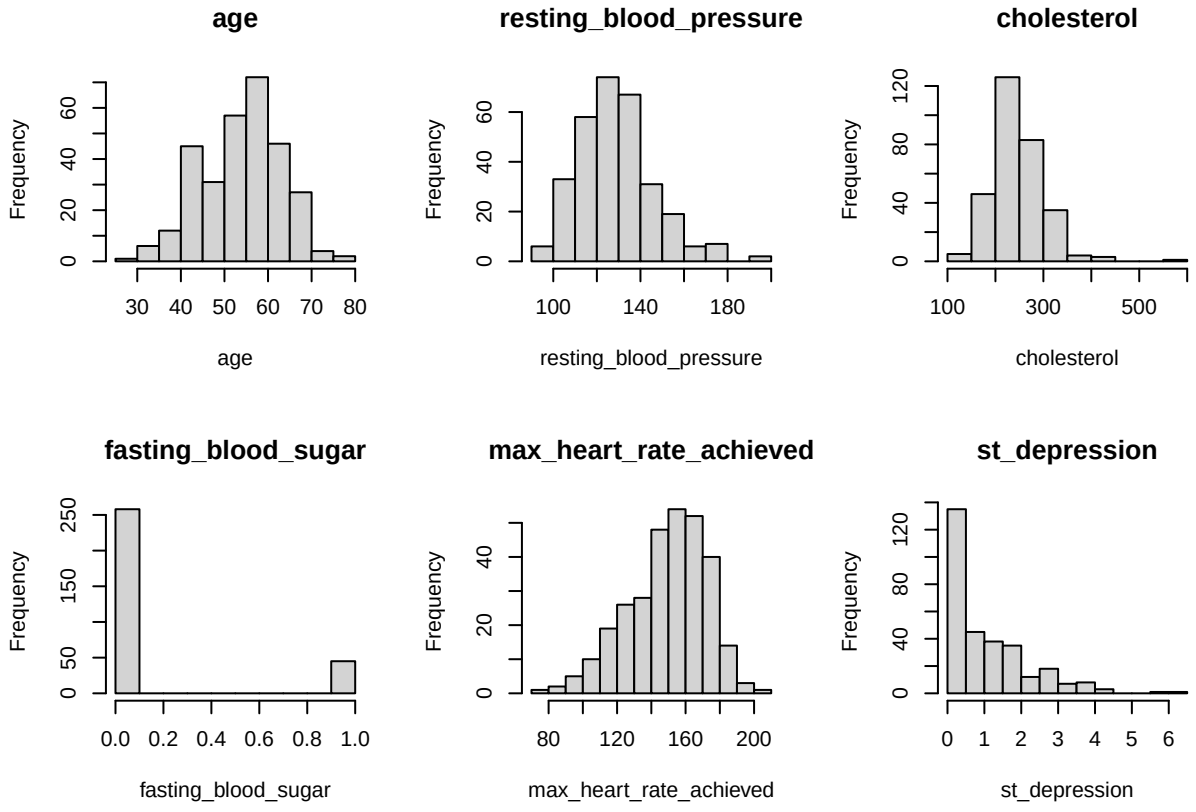
Por último, para nuestro análisis no se van a descartar registros porque no nos vamos a centrar en un tramo de edad concreto, sexo o una cantidad de colesterol, sino que se van a considerar a todos los pacientes con

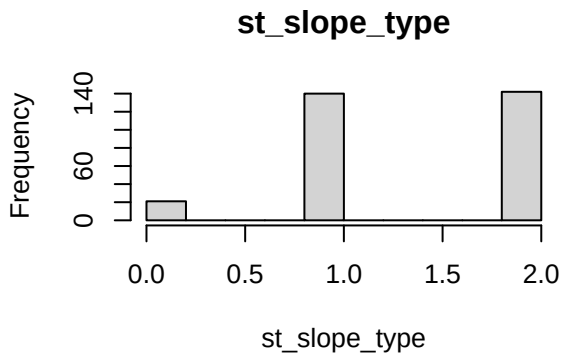
todas sus características para extraer el mayor número de conclusiones posibles teniendo en cuenta todos los atributos.

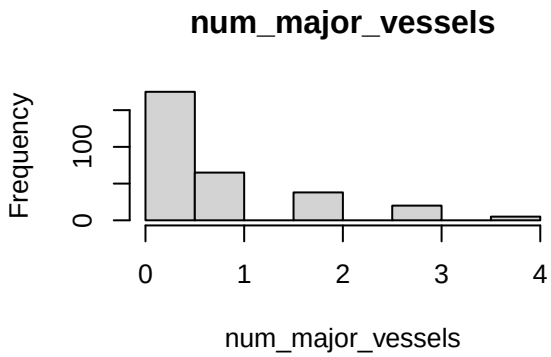
3 Visualización de la distribución de las variables

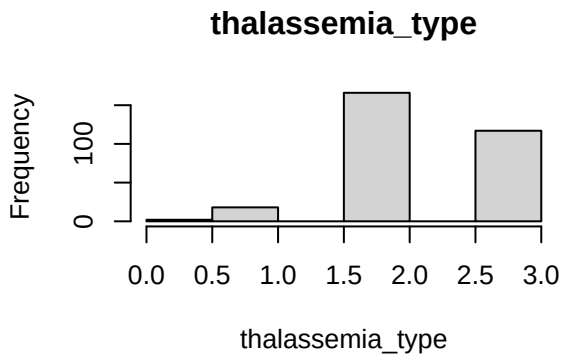
A continuación se lleva a cabo una visualización de datos que nos permite entender y analizar de forma gráfica el contenido del dataset. Existen diferentes tipos de gráficos que nos permiten visualizar la distribución de las variables de un dataset, como gráficos de barras, diagramas de cajas o histogramas. Para este caso se han utilizado histogramas, estos gráficos nos ayudan a comprender mejor la forma en que se distribuyen los valores de cada variable y a detectar posibles patrones o tendencias.

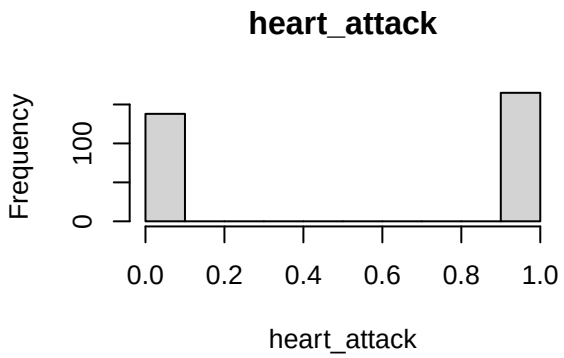
```
visualize_distribution <- function(variable) {
  # Seleccionamos la columna de la variable del conjunto de datos
  values <- heart_data[[variable]]
  # Creación del histograma
  hist(values, xlab = variable, main = variable)
}
```












```r
# Esta función recibe como argumento una variable 'x' y devuelve un vector con la media, la mediana y l
describe_variable <- function(x) {
  # Calcula la media de 'x' y la redondea a 3 decimales
  mean <- round(mean(x),3)
  # Calcula la mediana de 'x'
  median <- median(x)
  # Calcula la desviación típica de 'x' y la redondea a 3 decimales
  sd <- round(sd(x),3)
  # Crea un vector con la media, la mediana y la desviación típica
  result <- c(mean, median, sd)
  # Asigna nombres a los elementos del vector
  names(result) <- c("Media", "Mediana", "Desviación típica")
  # Devuelve el vector resultado
  return(result)
}
```

```r
# Ejecutamos la función con las variables
describe_variable(heart_data$age)
```

```
##             Media           Mediana Desviación típica
##            54.366            55.000             9.082
```

```r
describe_variable(heart_data$resting_blood_pressure)
```

```
##             Media           Mediana Desviación típica
```

```
##           131.624           130.000            17.538
```

```
describe_variable(heart_data$cholesterol)
```

```
##             Media           Mediana Desviación típica
##           246.264           240.000            51.831
```

```
describe_variable(heart_data$fasting_blood_sugar)
```

```
##             Media           Mediana Desviación típica
##             0.149             0.000             0.356
```

```
describe_variable(heart_data$max_heart_rate_achieved)
```

```
##             Media           Mediana Desviación típica
##           149.647           153.000            22.905
```

```
describe_variable(heart_data$st_depression)
```

```
##             Media           Mediana Desviación típica
##             1.040             0.800             1.161
```

```
describe_variable(heart_data$st_slope_type)
```

```
##             Media           Mediana Desviación típica
##             1.399             1.000             0.616
```

```
describe_variable(heart_data$num_major_vessels)
```

```
##             Media           Mediana Desviación típica
##             0.729             0.000             1.023
```

```
describe_variable(heart_data$thalassemia_type)
```

```
##             Media           Mediana Desviación típica
##             2.314             2.000             0.612
```

```
describe_variable(heart_data$heart_attack)
```

```
##             Media           Mediana Desviación típica
##             0.545             1.000             0.499
```

A partir de estos datos, se pueden obtener algunas conclusiones sobre las variables del dataset:

- La edad media de los pacientes es de 54.366 años, con una mediana de 55 años y una desviación típica de 9.082. Esto indica que la mayoría de los pacientes tienen una edad cercana a los 55 años, pero hay algunos pacientes más jóvenes y otros más mayores.

- La presión arterial media de reposo de los pacientes es de 131.624, con una mediana de 130 y una desviación típica de 17.538. Esto indica que la mayoría de los pacientes tienen una presión arterial cercana a los 130, pero hay algunos pacientes con presión arterial más baja y otros con presión arterial más alta.

- El colesterol medio de los pacientes es de 246.264, con una mediana de 240 y una desviación típica de 51.831. Esto indica que la mayoría de los pacientes tienen un nivel de colesterol cercano a los 240, pero hay algunos pacientes con niveles de colesterol más bajos y otros con niveles de colesterol más altos.

- El azúcar en sangre en ayunas medio de los pacientes es de 0.149, con una mediana de 0 y una desviación típica de 0.356. Esto indica que la mayoría de los pacientes tienen un nivel de azúcar en sangre en ayunas cercano a 0, pero hay algunos pacientes con niveles de azúcar en sangre en ayunas más bajos y otros con niveles de azúcar en sangre en ayunas más altos.

- La frecuencia cardíaca máxima alcanzada durante el ejercicio medio de los pacientes es de 149.647, con una mediana de 153 y una desviación típica de 22.905. Esto indica que la mayoría de los pacientes tienen una frecuencia cardíaca máxima alcanzada durante el ejercicio cercana a los 153, pero hay algunos pacientes con frecuencias cardíacas máximas alcanzadas durante el ejercicio más bajas y otros con frecuencias cardíacas máximas alcanzadas durante el ejercicio más altas.

- En la variable "st_depression" se puede observar que la media de esta variable es 1.04, lo que indica que en promedio, la depresión durante el ejercicio es de 1.04 unidades. La mediana de esta variable es 0.8, lo que significa que la mitad de los valores de esta variable son inferiores a 0.8. Por último, la desviación típica de esta variable es 1.161, lo que indica que los valores de esta variable tienen una gran variabilidad, ya que se extienden en un rango de 1.161 unidades a partir de la media.

- La media de la variable "st_slope_type" es 1.399, lo que indica que en promedio, la inclinación del segmento ST durante el ejercicio es de 1.399 unidades. La mediana de esta variable es 1, lo que significa que la mitad de los valores de esta variable son iguales a 1. La desviación típica de esta variable es 0.616, lo que indica que los valores de esta variable tienen una moderada variabilidad, ya que se extienden en un rango de 0.616 unidades a partir de la media.

- La media de la variable "num_major_vessels" es 0.729, lo que indica que en promedio, hay 0.729 vasos principales coloreados por fluoroscopía. La mediana de esta variable es 0, lo que significa que la mitad de los valores de esta variable son iguales a 0. La desviación típica de esta variable es 1.023, lo que indica que los valores de esta variable tienen una gran variabilidad, ya que se extienden en un rango de 1.023 unidades a partir de la media.

- La media de la variable "thalassemia_type" es 2.314, lo que indica que en promedio, el tipo de trombocitopenia es 2.314. La mediana de esta variable es 2, lo que significa que la mitad de los valores de esta variable son iguales a 2. La desviación típica de esta variable es 0.612, lo que indica que los valores de esta variable tienen una moderada variabilidad, ya que se extienden en un rango de 0.612 unidades a partir de la media.

- En el caso de la variable "heart_attack", podemos observar que el 54.5% de los pacientes del dataset no han sufrido un ataque al corazón. La media de esta variable es de 0.545 y su mediana es de 1, lo que indica que la mayoría de los pacientes no han sufrido un ataque al corazón. Además, su desviación típica es de 0.499, lo que sugiere que hay una cierta variabilidad en los datos.

4 Limpieza de los datos.

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

4.1.1 Caso: Ceros

```
# Analisis de las columnas que contienen ceros en sus valores
cols_with_zeros <- which(apply(heart_data, 2, function(x) sum(x == 0)) > 0)
colnames(heart_data)[cols_with_zeros]
```

```
##  [1] "sex"                    "chest_pain_type"
##  [3] "fasting_blood_sugar"    "rest_ecg_type"
##  [5] "exercise_induced_angina" "st_depression"
##  [7] "st_slope_type"          "num_major_vessels"
##  [9] "thalassemia_type"       "heart_attack"
```

Las variables que contienen algún valor igual a cero son variables que esperan reflejar este valor tal y como se ha definido en el enunciado, por lo tanto no se va a realizar una limpieza de datos para este caso en particular. Véase a continuación las variables que aparecen con algún valor cero son las siguientes:

- "sex": Refleja el género de la persona (1 = masculino, 0 = femenino).
- "chest_pain_type": Identifica el tipo de dolor en el pecho (0 = angina típica, 1 = angina atípica, 2 = dolor no anginoso, 3 = asintomático).
- "fasting_blood_sugar": Indica si el nivel de azúcar en sangre en ayunas es mayor a 120 mg/dl (1 = verdadero, 0 = falso).
- "rest_ecg_type": 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).
- "exercise_induced_angina": Indica si la angina ha sido inducida por el ejercicio (1 = sí, 0 = no).
- "st_depression": Señala la depresión ST inducida por el ejercicio en relación con el descanso.
- "st_slope_type": Muestra la pendiente del segmento ST de ejercicio máximo (0 = inclinación hacia abajo, 1 = plano, 2 = inclinación hacia arriba).
- "num_major_vessels": Indica el número de buques principales (0, 1, 2, 3).
- "thalassemia_type": 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)).
- "heart_attack": Indica si el paciente sufre un ataque al corazón o no (0 = No, 1 = Sí).

4.1.2 Caso: Elementos Vacíos

A continuación se realiza la comprobación de si hay elementos vacíos en el dataset, para cada columna se realiza el conteo de elementos vacíos existentes.

```r
# Elementos vacíos de las variables del dataset
colSums(is.na(heart_data))
```

```
##                       age                       sex         chest_pain_type
##                         0                         0                       0
##  resting_blood_pressure               cholesterol     fasting_blood_sugar
##                         0                         0                       0
##           rest_ecg_type max_heart_rate_achieved exercise_induced_angina
##                         0                         0                       0
##           st_depression           st_slope_type       num_major_vessels
##                         0                         0                       0
##        thalassemia_type            heart_attack
##                         0                         0
```

Como se visualiza en los resultados anteriores no existen elementos vacíos en el conjunto de datos. Con ello, no será necesario realizar ningún procedimiento de limpieza de datos para valores vacíos de las variables del dataset.

4.1.3 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"))
heart_data$heart_attack <- factor(heart_data$heart_attack, levels = c(0,1),
                                      labels= c("No", "Yes"))
```

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.

```
# 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.

4.2 Identifica y gestiona los valores extremos

En primer lugar se realiza la visualización de los valores extremos para las variables: *"age", "cholesterol", "max_heart_rate_achieved", "resting_blood_pressure", "st_depression".*

```
outlier_info <- function(var, name_var, show_plot = TRUE) {
  # Valores extremos en formato boxplot de la variable
  if (show_plot) {
    boxplot(var, main = name_var,
            ylab="Valor", col = "lightblue", horizontal = FALSE, outline = TRUE)
  }

  # Identificar los valores atípicos
  outliers <- boxplot.stats(var)$out

  # Imprimir los valores máximo y mínimo de los valores atípicos
  stats <- boxplot.stats(var)$stats
  cat("Valor mínimo:", stats[1], "\n")
  cat("Primer cuartil:", stats[2], "\n")
  cat("Media:", stats[3], "\n")
  cat("Tercer cuartil:", stats[4], "\n")
  cat("Valor máximo:", stats[5], "\n")

  if (length(outliers) == 0) {
    cat("No se han identificado valores atípicos", "\n")
  } else {
    # Imprimir el número de valores atípicos
    cat("Outliers identificados:", unique(outliers), "\n")
  }
}
```

```
par(mfrow=c(2, 3))
outlier_info(heart_data$age, "age")
```

```
## Valor mínimo: 29
## Primer cuartil: 47.5
## Media: 55
## Tercer cuartil: 61
## Valor máximo: 77
## No se han identificado valores atípicos
```

```
outlier_info(heart_data$cholesterol, "cholesterol")
```

```
## Valor mínimo: 126
## Primer cuartil: 211
## Media: 240
## Tercer cuartil: 274.5
## Valor máximo: 360
## Outliers identificados: 417 564 394 407 409
```

```
outlier_info(heart_data$max_heart_rate_achieved, "max_heart_rate_achieved")
```

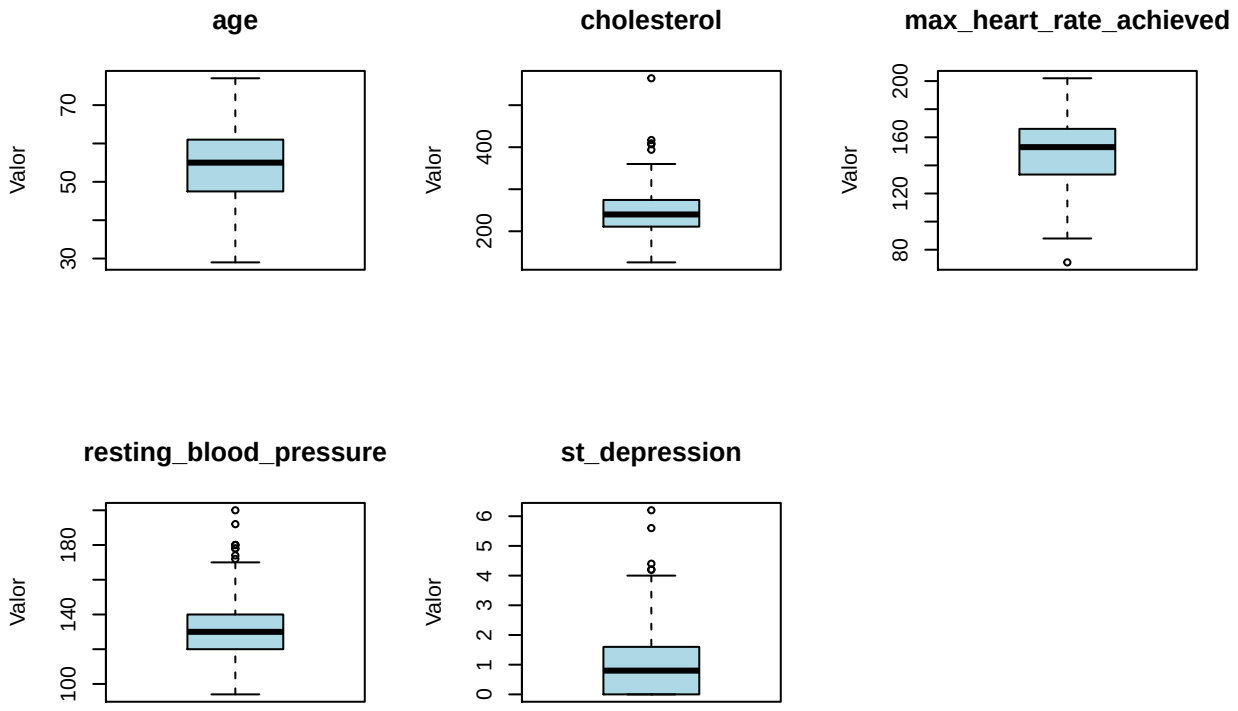
```
## Valor mínimo: 88
## Primer cuartil: 133.5
## Media: 153
## Tercer cuartil: 166
## Valor máximo: 202
## Outliers identificados: 71
```

```
outlier_info(heart_data$resting_blood_pressure, "resting_blood_pressure")
```

```
## Valor mínimo: 94
## Primer cuartil: 120
## Media: 130
## Tercer cuartil: 140
## Valor máximo: 170
## Outliers identificados: 172 178 180 200 174 192
```

```
outlier_info(heart_data$st_depression, "st_depression")
```

```
## Valor mínimo: 0
## Primer cuartil: 0
## Media: 0.8
## Tercer cuartil: 1.6
## Valor máximo: 4
## Outliers identificados: 4.2 6.2 5.6 4.4
```



A continuación vamos a extraer las conclusiones pertinentes respectos a los valores extremos detectados en los resultados y los gráficos previos:

- En la variable "age", no se han identificado valores atípicos. Los valores máximo y mínimo de la variable son 29 y 77, respectivamente.
- En la variable "cholesterol", se han identificado 5 valores atípicos (outliers). Los valores máximo y mínimo de los outliers identificados son 126 y 564, respectivamente.
- En la variable "max_heart_rate_achieved", se ha identificado 1 valor atípico (outlier). Los valores máximo y mínimo de los outliers identificados son 71 y 202, respectivamente.
- En la variable "resting_blood_pressure", se han identificado 6 valores atípicos (outliers). Los valores máximo y mínimo de los outliers identificados son 94 y 200, respectivamente.
- En la variable "st_depression", se han identificado 4 valores atípicos (outliers). Los valores máximo y mínimo de los outliers identificados son 0 y 6.2, respectivamente.

Estos resultados indican que algunas de las variables tienen valores extremos que se alejan significativamente del resto y que pueden afectar el rendimiento de algunos algoritmos de análisis de datos.

4.3 Corrección de los outliers

Se van a tratar los valores de outliers que hemos considerado como no válidos. Es importante destacar que un valor extremo no tiene por que ser no válido, para determinar si un valor extremo es válido o no hemos realizado una investigación sobre las variables y los posibles valores que estás pueden tener. De todos los outliers detectados simplemente nos centramos en el caso de la variable *cholesterol*.

Realizando una búsqueda sobre los valores comúnes y menos comúnes de cholesterol (mg / dL) en 300 mg/dl o más ya se considera un nivel muy alto. Para casos más elevados se habla de sufrir hipertrigliceridemia. Nosotros hemos establecido que para un valor mayor a 550 se va a realizar una corrección de este valor.

```
# Número de outliers que superan el valor establecido en la variable tratada
num_outliers_var = nrow(heart_data[heart_data$cholesterol >550,])

# Número de NA's en la variable tratada
num_nas_variable_inicio = sum(is.na(heart_data$cholesterol))

# Se substituyen esos valores atipicos por el valor NA
heart_data = heart_data %>% mutate(cholesterol = ifelse(cholesterol > 550, NA, cholesterol))

# Numero de NA's final después del tratado en la variable
num_nas_variable_final = sum(is.na(heart_data$cholesterol))

# Visualización de los resultados
print(paste("Número de outliers que superan el valor establecido en la variable:", num_outliers_var))
```

```
## [1] "Número de outliers que superan el valor establecido en la variable: 1"
```

```
print(paste("Número inicial de NA's en la variable:", num_nas_variable_inicio))
```

```
## [1] "Número inicial de NA’s en la variable: 0"
```

```
print(paste("Número final de NA's en la variable después del tratado:", num_nas_variable_final))
```

```
## [1] "Número final de NA’s en la variable después del tratado: 1"
```

4.4 Imputación de valores mediante kNN

A continuación se va a realizar una imputación de valores perdidos. Aplicaremos imputación por vecinos más cercanos, utilizando la distancia de Gower, considerando en el cómputo de los vecinos más cercanos el resto de variables cuantitativas.

Para realizar esta imputación, usaremos la función “kNN” de la librería VIM con un número de vecinos igual a 11. Se mostrarña que la imputación se ha realizado correctamente, mostrando el resultado de los datos afectados por la imputación.

```
# Selección de los datos a imputar
df_auxiliar <- heart_data %>%
  select(cholesterol, fasting_blood_sugar, rest_ecg_type, resting_blood_pressure, st_depression) %>%
  filter((is.na(cholesterol)))
```

```
# Se aplica imputación por vecinos mas cercanos (KNN)
# Se imputan los registros de género masculino
df_auxiliar_knn <- kNN(df_auxiliar,
                       variable = c("cholesterol"),
                       k = 11,
                       dist_var = c("rest_ecg_type", "resting_blood_pressure", "st_depression"))
```

```r
# Ahora se eliminarán del dataset original los registros de los dataframes auxiliares
# Dimensión actual del dataframe original
dim(heart_data)
```

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

```r
# Se elimina del dataset original los registros que ya están imputados
heart_data = heart_data %>% filter(!(is.na(cholesterol)))
# Dimension del dataframe original
dim(heart_data)
```

```
## [1] 302  14
```

4.5 Generación del archivo con los datos tratados

Se genera el fichero con los datos tratados y limpiados tal y como se pide en la práctica.

```r
# Dataframe tratado
df_heart_final <- heart_data
# Se incluyen las variables cuantitativas normalizadas
df_heart_final[, idx_var_cuant] <- heart_norm
# Se exporta a formato csv
write.csv(df_heart_final, file = "clean_data.csv", row.names = FALSE, col.names = TRUE)
```

5 Análisis de los datos.

5.1 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?).

==== JOAN ====

En este caso, se quiere analizar el conjunto de datos "heart.csv" para predecir si un paciente tiene un ataque al corazón o no. El conjunto de datos "o2Saturation.csv" no parece estar relacionado con este propósito y, por tanto, no se incluiría en el análisis.

Para analizar el conjunto de datos "heart.csv", se podría utilizar un modelo de aprendizaje supervisado para entrenar un modelo que use las variables independientes (edad, género, tipo de dolor en el pecho, etc.) como entrada y la variable dependiente (ataque al corazón o no) como salida. Una vez entrenado el modelo, se puede evaluar su desempeño y utilizarlo para hacer predicciones sobre pacientes futuros.

También se pueden comparar los resultados de distintos modelos de aprendizaje supervisado para ver cuál tiene mejor desempeño en este problema en particular. También se puede comparar el desempeño del modelo entrenado con diferentes subconjuntos de datos (por ejemplo, separando los pacientes por género o por edad).

Se podría utilizar un árbol de decisión para construir un modelo que use las variables independientes (edad, género, tipo de dolor en el pecho, etc.) como entrada y la variable dependiente (ataque al corazón o no) como salida. Una vez entrenado el modelo, se podría utilizar para hacer predicciones sobre pacientes futuros.

Para evaluar el desempeño del modelo, se podrían utilizar métricas como la precisión, la sensibilidad o el valor F1. También se podría comparar el desempeño del árbol de decisión con otros modelos de aprendizaje supervisado para ver cuál tiene mejor desempeño en este problema en particular.

Además, podríamos utilizar el test Chi cuadrado para ver si existe alguna asociación entre el género de un paciente y el resultado (ataque al corazón o no).

Para utilizar el test Chi cuadrado, necesitaríamos crear una tabla de contingencia con las frecuencias absolutas o relativas de cada combinación de variables. Luego, se calcularía el valor Chi cuadrado y se compararía con una tabla de valores críticos para determinar si existe una asociación significativa entre las variables.

==== LUCAS ====

Como comentamos al principio de la práctica, queremos responder a las siguientes preguntas:

- ¿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?
- ¿Tuvieron los pacientes con una angina de pecho producida por el ejercicio físico más probabilidad de sufrir un ataque que los que no?
- ¿Las personas mayores tienen más probabilidad de sufrir un ataque que los demás?
- ¿ Hubo algún indicio de sufrir más fácilmente un ataque al corazón según el dolor de pecho del paciente ?

En nuestro caso, se va a analizar el conjunto de datos "heart.csv" para poder predecir si un paciente tiene un ataque al corazón o no. Para ello, se podría utilizar un modelo de aprendizaje supervisado como un árbol de decisión para construir un modelo que use las variables independientes (edad, género, tipo de dolor en el pecho, etc.) como entrada y la variable dependiente (ataque al corazón o no) como salida. Una vez entrenado el modelo, se puede evaluar su desempeño con métricas como la precisión, la sensibilidad o el valor F1 y utilizarlo para hacer predicciones sobre pacientes futuros.

Además, se van a aplicar algunos tests estadísticos que dependerá de la normalidad y la homocedasticidad de las variables que se verá en el apartado siguiente.

5.2 Comprobación de la normalidad y homogeneidad de la varianza.

Se va a analizar la normalidad y la homocedasticidad de las variables cuantitativas que nos servirán para dar respuesta a las preguntas anteriores.

Para ello, vamos a representar mediante histogramas la distribución de los datos de las variables en comparación con la normal teórica para poder ver visualmente si siguen una distribución normal. No obstante, después lo verificaremos mediante el test de *Shapiro Wilk* y mediante el gráfico *Q-Q plot* mediante las funciones de R `qqnorm` y `qqline`.

```
# Se carga la libreria psych
#library(psych)

# Histograma de la distribución de la variable VS la distribucion normal teorica
#multi.hist(x = heart_data[,idx_var_cuant], dcol = c("blue", "red"), dlty = c("dotted", "solid"),global
```

Podemos comprobar como visualmente no siguen una distribución normal ninguna de las variables, sin embargo hay algunas como

```
# Devuelve el p-valor aplicando el test de Shapiro Wilk
p_value_shapiro_wilk <- function(x) {
  p_value <- shapiro.test(x)["p.value"]
  return (p_value)
```

```
}

# Se crea una dataframe con los p-valores obtenidos para cada variable
df_p_value_shapiro_wilk <- data.frame(
"P-Value" = sapply(heart_data[,idx_var_cuant],p_value_shapiro_wilk))

# Se le añade el nombre a las variables
colnames(df_p_value_shapiro_wilk) <- c("Age","Resting_Blood_Pressure","Cholesterol","Max_heart_rate_achi

# Se visualiza el dataframe creado
kable(df_p_value_shapiro_wilk,digits=3, caption="P-Valores de las variables cuantitativas aplicando Shap
```

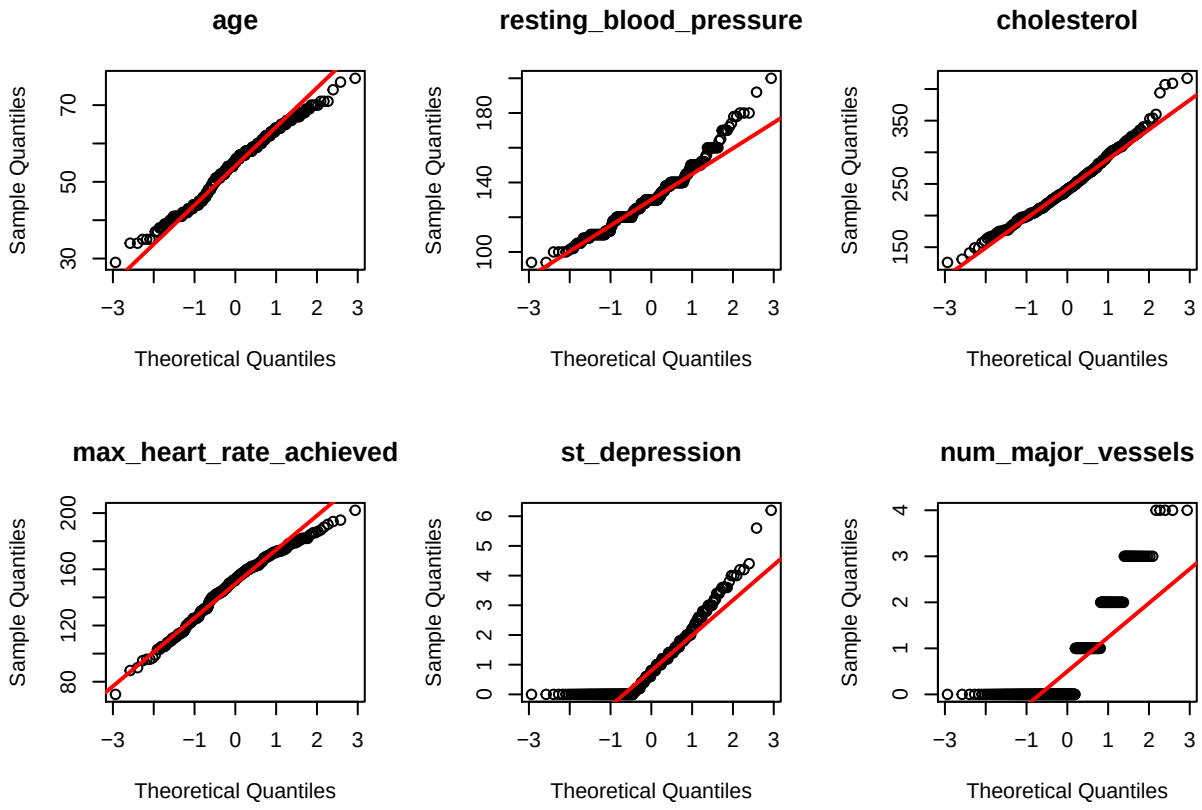
Table 1: P-Valores de las variables cuantitativas aplicando Shapiro Wilk

Age	Resting_Blood_Pressure	Cholesterol	Max_heart_rate_achieved	t_depression	num_major_vessels
0.007	0	0.001	0	0	0

Viendo los resultados del test se rechaza la hipótesis nula y se confirma que la distribución de las variables no siguen una distribución normal al 95% de confianza ya que el p-valor obtenido en cada caso es inferior al nivel de significancia del 5 % (0.05).

Por último, se muestra el *Q-Q plot* para comprobar si los cuantiles siguen o no una distribución normal.

```
par(mfrow = c(2, 3))
for (var in idx_var_cuant){
  qqnorm(heart_data[,var], main=colnames(heart_data)[var], pch=1)
  qqline(heart_data[,var],col='red', lwd=2) }
```



!#TODO: ESTO ES UN EJEMPLO DE PARTIDA

Para comprobar la normalidad de los datos, utilizamos la función `shapiro.test()`. Esta función toma un vector de datos y realiza un test de normalidad de Shapiro-Wilk. Si el p-valor devuelto es superior al nivel de significación, entonces se puede concluir que los datos siguen una distribución normal.

```
# Carga el conjunto de datos
data(iris)

# Extrae la longitud del conjunto de datos y realiza un test de normalidad
shapiro.test(iris$Sepal.Length)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  iris$Sepal.Length
## W = 0.97609, p-value = 0.01018
```

Para comprobar la homogeneidad de la varianza, utilizamos la función `var.test()`. Esta función toma dos vectores de datos y realiza un test de igualdad de varianzas. Si el p-valor devuelto es superior al nivel de significación que hayas elegido, entonces se puede concluir que las varianzas de los dos conjuntos de datos son iguales.

```
# Carga el conjunto de datos
data(iris)
```

```r
# Compara la varianza
var.test(iris$Sepal.Length, iris$Petal.Length)
```

```
##
##  F test to compare two variances
##
## data:  iris$Sepal.Length and iris$Petal.Length
## F = 0.22004, num df = 149, denom df = 149, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1594015 0.3037352
## sample estimates:
## ratio of variances
##          0.2200361
```

5.3 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.

6 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.

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

8 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.

9 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.