


Practica_limpieza_y_analisis_de_datos

2023-02-01

Contents

1. Detalles de la actividad	3
1.1. Descripción	3
1.2. Objetivos	3
1.3. Competencias	3
2.Resolución	4
2.1. Descripción del dataset	4
2.2. Importancia y objetivos de los análisis	5
2.3. Limpieza de los datos	5
2.4. Análisis de los datos	12
2.5. Pruebas estadísticas	17
2.6.Conclusiones	20
3.Referencia	22



Tipología y ciclo de vida de los datos

aula 1

PRACTICA 2

Universidad Oberta de Catalunya
Creado por: Oscar Augusto Díaz Triana



1. Detalles de la actividad

1.1. Descripción

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

1.2. Objetivos

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

1.3. Competencias

- Capacidad para extraer, interpretar y analizar datos de diferentes entornos.
- Uso avanzado de las herramientas de programas estadísticos adecuados para los diferentes problemas de modelización, análisis y visualización de datos.
- Capacidad para proponer soluciones innovadoras y tomar decisiones.
- Capacidad para trabajar en equipos multidisciplinarios.

2.Resolución

2.1. Descripción del dataset

El conjunto de datos objeto de análisis se ha obtenido a partir de este enlace en Kaggle y está constituido por 14 características (columnas) que presentan 303 registros de personas (filas).

Entre los campos de este conjunto de datos, encontramos los siguientes:

- **age** : Edad del paciente
- **sex** : Género: 1 = masculino; 0 = femenino
- **cp** : Tipo de dolor en el pecho (valores de 0 a 3)
 - Valor 0: angina típica
 - Valor 1: angina atípica
 - Valor 2: dolor no anginoso
 - Valor 3: 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 (1 = verdadero; 0 = falso)
- **restecg**: Resultados electrocardiográficos en reposo (Valores de 0 a 2)
 - Valor 0: Normal
 - Valor 1: Anomalía de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST de > 0,05 mV)
 - Valor 2: Muestra hipertrofia ventricular izquierda probable o definida según el criterio de Estes
- **thalachh** : Máxima frecuencia cardíaca alcanzada
- **exng** : 0= menos probabilidad de ataque cardíaco 1= más probabilidad de ataque cardíaco
- **oldpeak**: Depresión del ST inducida por el ejercicio relativo al descanso.
- **slp**: Pendiente de ST en el pico de ejercicio (Del 0 al 2).
 - Valor 0: Ascendente
 - Valor 1: Plana
 - Valor 2: Descendente
- **caa**: Número de vasos principales (0-4) coloreados por fluorospía.
- **thall** : THAL. (Del 0 al 3). Esta escala no logre identificarla a que equivale cada valor
 - Valor 0:
 - Valor 1:
 - Valor 2:
 - Valor 3:
- **Output**: Variable de destino 0 = No tiene enfermedad cardiovascular. 1 = Tiene enfermedad cardiovascular.

```
#cargue de las librerías que vamos a utilizar
library(Rcmdr) #trabajar con R commander
```

```
## Loading required package: splines
## Loading required package: RcmdrMisc
## Loading required package: car
## Loading required package: carData
## Loading required package: sandwich
## Loading required package: effects
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.

## La interfaz R-Commander sólo funciona en sesiones interactivas

##
## Attaching package: 'Rcmdr'

## The following object is masked from 'package:base':
##
##      errorCondition

library(ggplot2) # permite trabajar graficar, otra opción
library(lattice) # permite trabajar factores
library(moments) # permite trabajar factores
library(splines) # lebreria complementaria a las primeras
library(RcmdrMisc) # lebreria complementaria a las primeras
library(car) # lebreria complementaria a las primeras
library(carData) # lebreria complementaria a las primeras
library(sandwich) # lebreria complementaria a las primeras
```

2.2. Importancia y objetivos de los análisis

A partir de este conjunto de datos se plantea la problemática de determinar qué variables influyen más sobre determinar que personas pueden sufrir ataques cardíacos.

Además, se podrá proceder a crear modelos de regresión que permitan predecir que personas pueden sufrir ataques cardíacos en función de sus características y contrastes de hipótesis que ayuden a identificar propiedades interesantes en las muestras que puedan ser inferidas con respecto a la población.

Estos análisis adquieren una gran relevancia en casi cualquier sector relacionado con la salud, ya que con base a este diagnóstico los médicos especialistas de las diferentes áreas del cuerpo podrán atender y tratar a las personas para que no le genera complicaciones cardíacas al paciente. Igualmente conociendo este diagnóstico se pueden generar estrategias para disminuir el riesgo de un ataque cardíaco modificando conductas que pueden aumentar el riesgo o tratando una enfermedad coronaria conocida. Los cambios para un estilo de vida saludable, que incluyen comer alimentos saludables, mantenerse activo, dejar de fumar, controlar el estrés y mantener un peso saludable, pueden ayudar a prevenir la enfermedad cardíaca. Incluso si ya tiene enfermedad coronaria, esos cambios pueden disminuir el riesgo de un ataque cardíaco.

2.3. Limpieza de los datos

Antes de comenzar con la limpieza de los datos, procedemos a realizar la lectura del fichero en formato CSV en el que se encuentran los datos. El resultado devuelto por la llamada a la función `read.csv()` será un objeto `data.frame`:

```
# Lectura de datos
corazon <- read.csv("heart.csv", header = TRUE)
head(corazon[,1:5])
```

```
##   age sex cp trtbps chol
## 1  63  1  3   145  233
## 2  37  1  2   130  250
## 3  41  0  1   130  204
## 4  56  1  1   120  236
## 5  57  0  0   120  354
## 6  57  1  0   140  192
```

```
# Tipo de dato asignado a cada campo
sapply(corazon, function(x) class(x))
```

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

Además, observamos cómo los tipos de datos asignados automáticamente por R a las variables se corresponden con el dominio de estas.

Nota: se tenía previsto que los valores desconocidos eran denotados en el dataset mediante el carácter ‘?’. y se pretendía realizar una sustitución de estos valores por una cadena vacía previa a la lectura, para que R marque estos valores desconocidos como NA (del inglés, Not Available). Esto simplificaría el manejo de los datos en los apartados posteriores, sin embargo el data set no contenía valores desconocidos.

2.3.1. Selección de los datos de interés

La gran mayoría de los atributos presentes en el conjunto de datos se corresponden con características que se reúnen de las diferentes personas y que fueron recogidos en forma de registros, por lo que será conveniente tenerlos en consideración durante la realización de los análisis. No se encontró características de las cuales se pudieran prescindir de ellas, como para ser eliminadas.

2.3.2. Ceros y elementos vacíos

Comúnmente, se utilizan los ceros como centinela para indicar la ausencia de ciertos valores. Sin embargo, no es el caso de este conjunto de datos puesto que, como se comentó durante el apartado relativo a la lectura, no se encontraron valores desconocidos. Así, se procede a verificar si los campos contienen elementos vacíos:

```
# Números de valores desconocidos por campo
sapply(corazon, function(x) sum(is.na(x)))
```

```
##      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
```

Si fuese el caso de encontrar valores desconocidos, deberíamos decidir cómo manejar estos registros que contienen valores desconocidos para algún campo. Una opción podría ser eliminar esos registros que incluyen este tipo de valores, pero ello supondría desaprovechar información.

Como alternativa, se empleará un método de imputación de valores basado en la similitud o diferencia entre los registros: la imputación basada en k vecinos más próximos (en inglés, kNN-imputation). La elección de esta alternativa se realiza bajo la hipótesis de que nuestros registros guardan cierta relación. No obstante, es mejor trabajar con datos “aproximados” que con los propios elementos vacíos, ya que obtendremos análisis con menor margen de error.

A continuación, se realiza el ejemplo de cómo se debe realizar la imputación si hubiese campos vacíos en todas las características.

Imputación de valores mediante la función kNN() del paquete VIM, no aparecerá el mensaje de nada que i

```
suppressWarnings(suppressMessages(library(VIM)))
```

```
corazon$age <- kNN(corazon)$age
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$sex <- kNN(corazon)$sex
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$cp <- kNN(corazon)$cp
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$trtbps <- kNN(corazon)$trtbps
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$chol <- kNN(corazon)$chol
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$fbs <- kNN(corazon)$chol
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$restecg <- kNN(corazon)$restecg
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$thalachh <- kNN(corazon)$thalachh
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$exng <- kNN(corazon)$exng
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$oldpeak <- kNN(corazon)$oldpeak
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$slp <- kNN(corazon)$slp
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also  
## after using makeNA)
```

```
corazon$caa <- kNN(corazon)$caa
```

```
## Warning in kNN(corazon): Nothing to impute, because no NA are present (also
```

```
## after using makeNA)
corazon$thall <- kNN(corazon)$thall

## Warning in kNN(corazon): Nothing to impute, because no NA are present (also
## after using makeNA)
corazon$output <- kNN(corazon)$output

## Warning in kNN(corazon): Nothing to impute, because no NA are present (also
## after using makeNA)
sapply(corazon, function(x) sum(is.na(x)))

##      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
```

2.3.3. Valores extremos

Los valores extremos o outliers son aquellos que parecen no ser congruentes sin los comparamos con el resto de los datos. Para identificarlos, podemos hacer uso de dos vías: (1) representar un diagrama de caja por cada variable y ver qué valores distan mucho del rango intercuartílico (la caja) o (2) utilizar la función `boxplots.stats()` de R. Se utilizaran las 2 opciones:

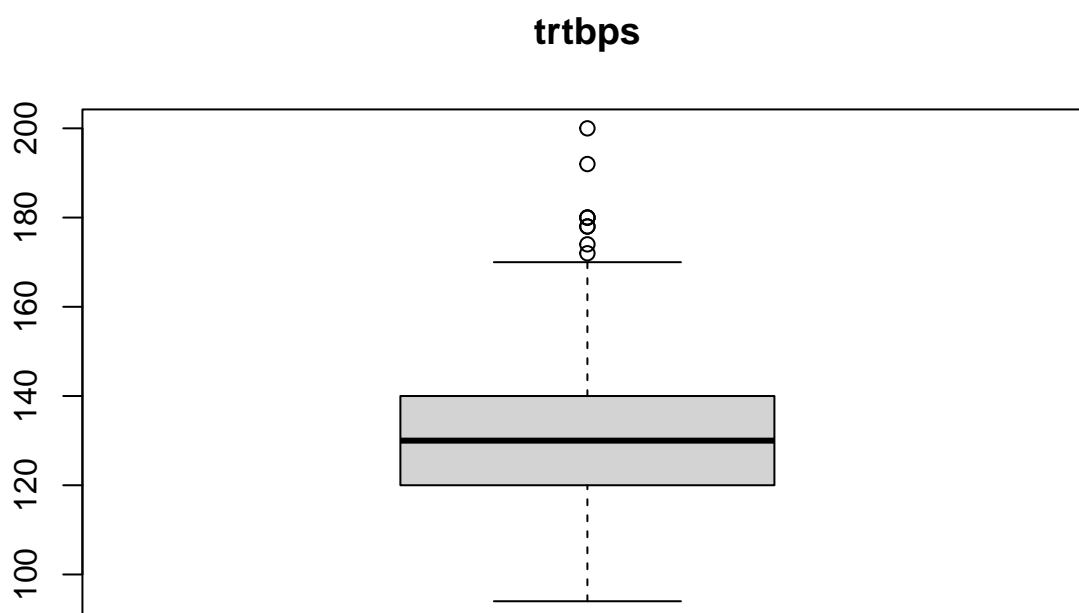
A continuación se mostrarán sólo los valores atípicos para aquellas variables que los contienen:

Variable presión arterial en reposo

```
# Valores atípicos para la variable trtbps
boxplot.stats(corazon$trtbps)$out

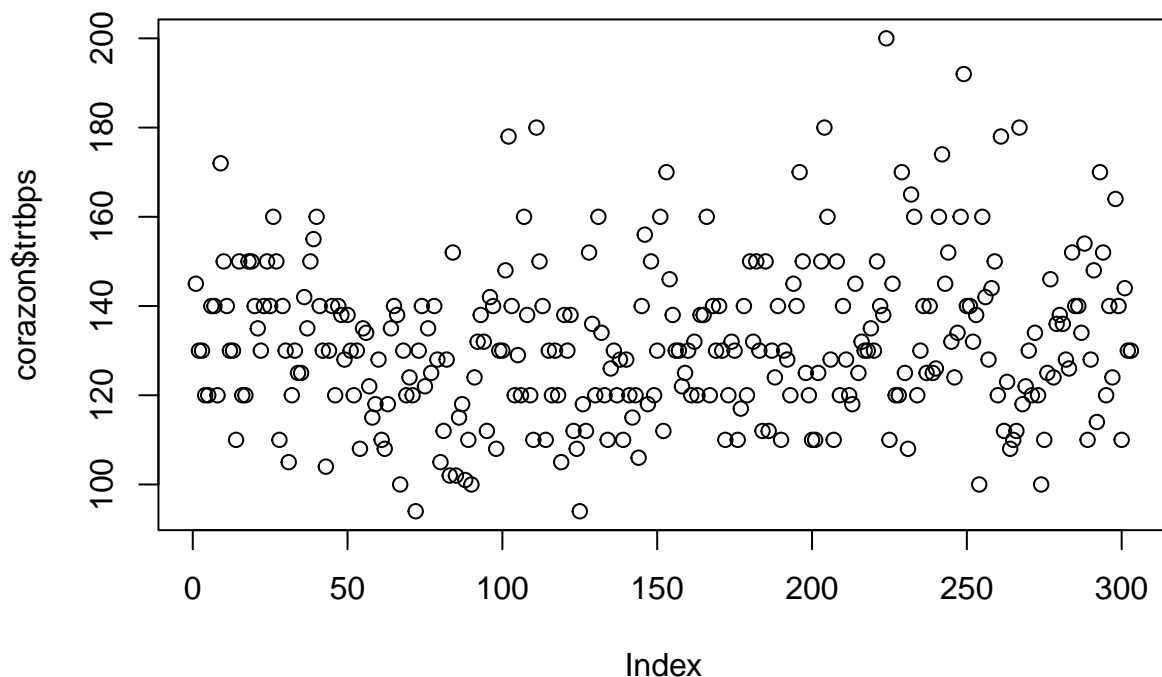
## [1] 172 178 180 180 200 174 192 178 180

# Diagrama de caja para la variable trtbps
boxplot(corazon$trtbps)
title("trtbps")
```

En la presión arterial en reposo, los valores fuera y alejados de la caja no se tratan de valores extremos que se traduzcan en error de inserción. Si lo vemos en un gráfico plot, se entiende mejor.

```
plot(corazon$trtbps)
```



```
#title("Presión arterial en reposo")
```

En este estudio nos interesa encontrar esos picos en los niveles de presión arterial, ya que se salen del rango recomendable, y lo mismo si fueran demasiado bajos. para esta variable la máxima frecuencia cardíaca alcanzada, podemos determinar que las presión maxima de los pacientes puede estar entre 172 y 200. Sin embargo el manejo de estos valores extremos consistirá en simplemente dejarlos como actualmente están los registros, ya que corresponden a las tomas realizadas a los pacientes.

Variable colesterol

```
# Valores atípicos para la variable chol
boxplot.stats(corazon$chol)$out
```

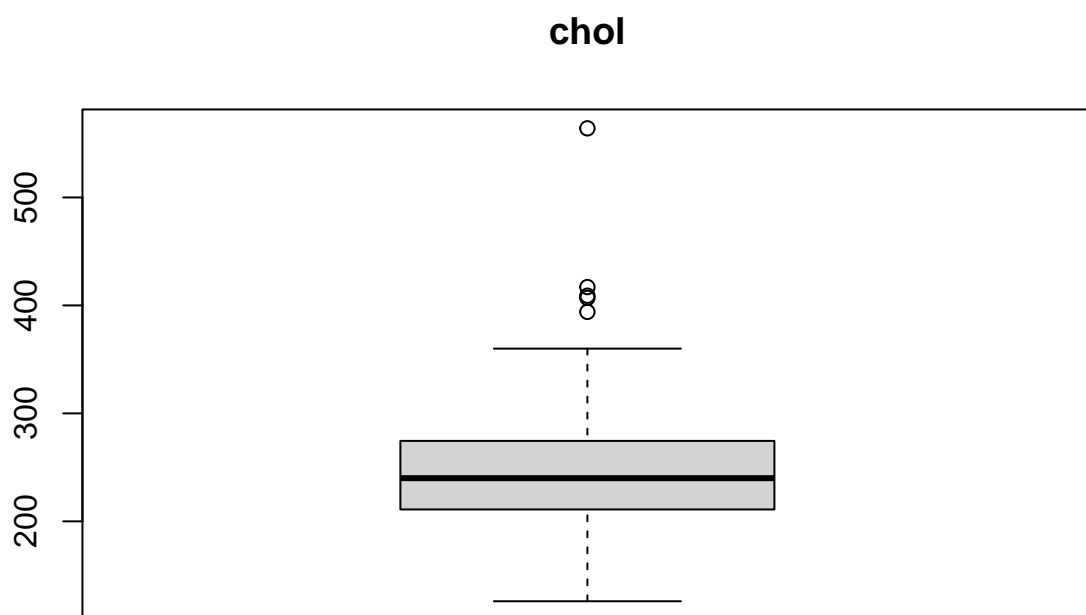
```
## [1] 417 564 394 407 409
```

En los niveles de colesterol encontramos un caso donde el valor de estos son 564. Aunque es un valor muy alto, tiene sentido ya que esta persona indica enfermedad (valor output = 1) osea que tiene enfermedad cardiovascular.

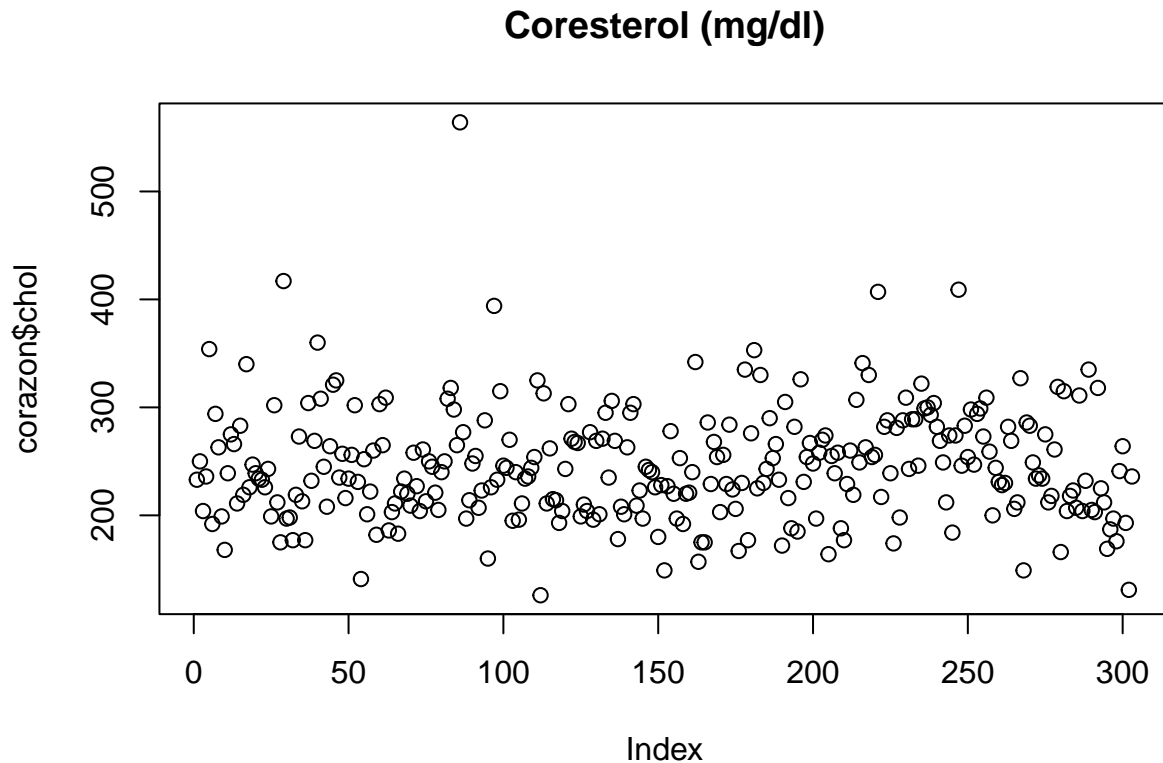
```
corazon[which(corazon$chol>500),]
```

```
##   age sex cp trtbps chol fbs restecg thalachh exng oldpeak slp caa thall
## 86  67  0  2   115  564  564      0     160    0     1.6   1   0     3
##   output
## 86      1
```

```
# Diagrama de caja para la variable chol
boxplot(corazon$chol)
title("chol")
```



```
plot(corazon$chol)
title("Coresterol (mg/dl)")
```



2.3.4. Exportación de los datos preprocesados

Después de haber aplicado sobre el conjunto de datos inicial los procedimientos de integración, validación y limpieza anteriores, procedemos a guardar estos en un nuevo fichero denominado `corazon_data_clean.csv`:

```
# Exportación de los datos limpios en .csv
write.csv(corazon, "corazon_data_clean.csv")
```

2.4. Análisis de los datos

2.4.1. Selección de los grupos de datos a analizar

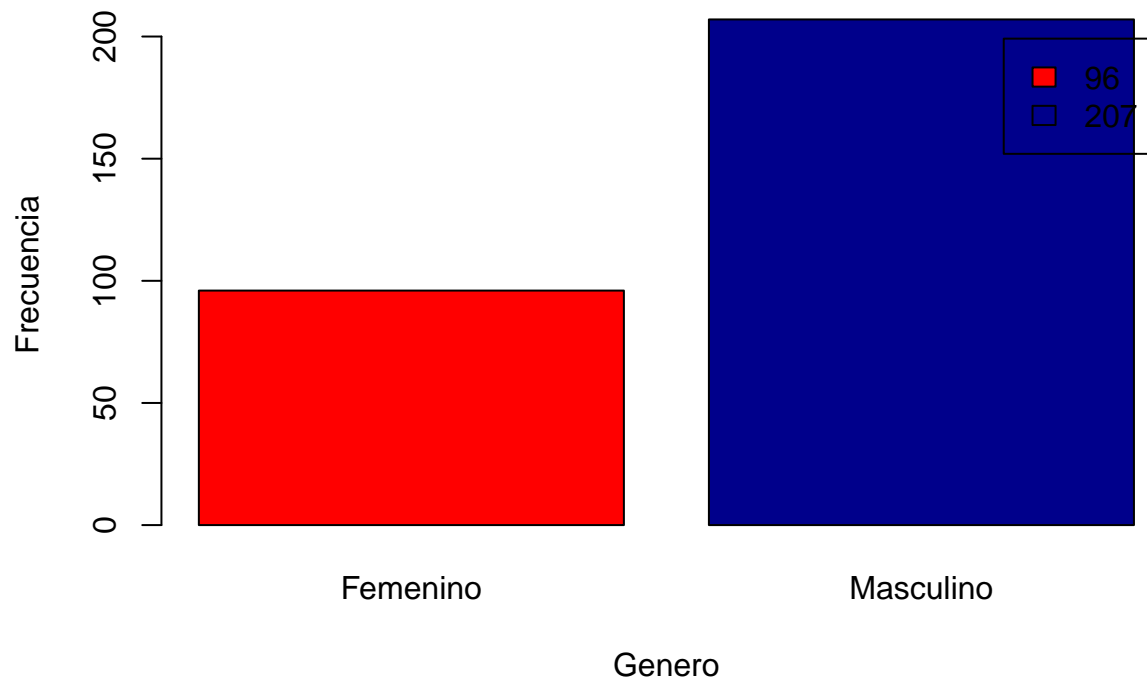
A continuación, se seleccionan los grupos dentro de nuestro conjunto de datos que pueden resultar interesantes para analizar y/o comparar. No obstante, como se verá en el apartado consistente en la realización de pruebas estadísticas, no todos se utilizarán.

```
# Agrupación por tipo de combustible
corazon.femenino <- corazon[corazon$sex == "0",]
corazon.masculino <- corazon[corazon$sex == "1",]
```

Para la selección de los grupos de datos a analizar, también se puede apoyar realizando la relación que pueda existir entre las variables

```
corazon.numeric <- corazon[,unlist(lapply(corazon, is.numeric))]
plot(corazon.numeric)
```

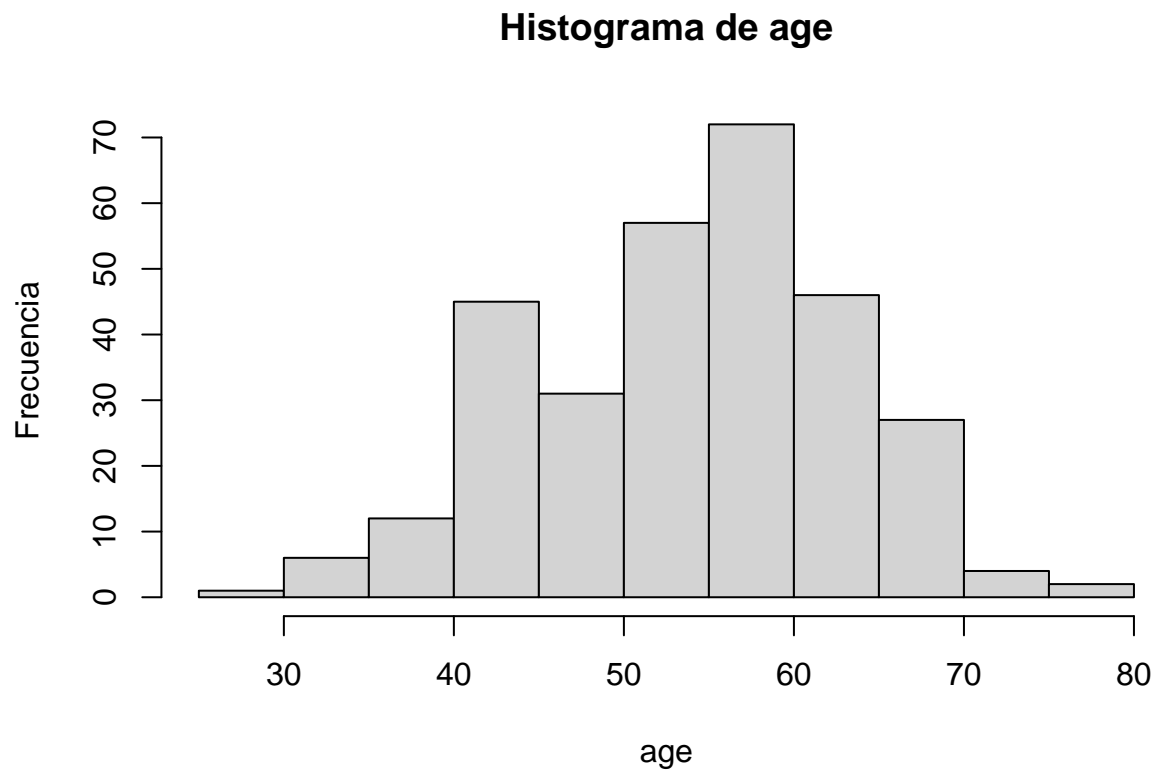

Gráfico de barras variable sex



Vamos de el data set contine mas registros de hombre que mujeres, se podria suponer que los hombres son las propensos a tener enfermedades cordiovasculaes

Seguidamente se realiza un histograma para conocer la distribución de la variable edad .

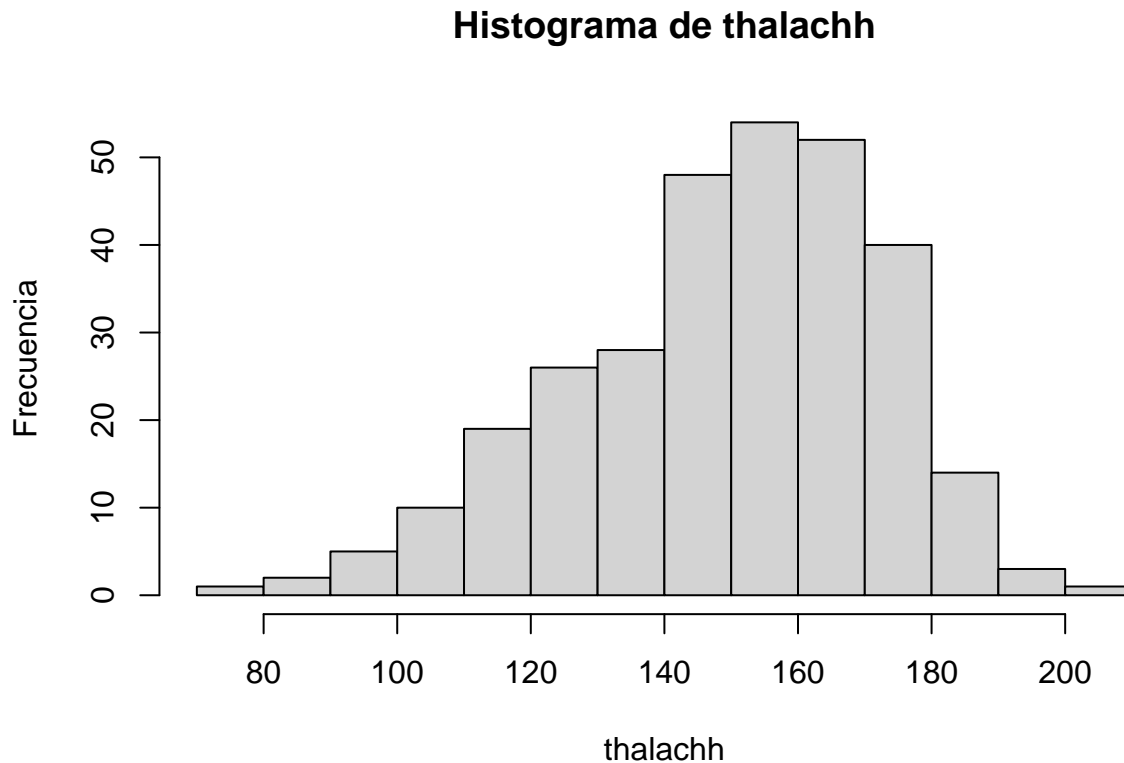
```
hist(corazon$age, main = "Histograma de age", # Frecuencia  
      xlab = "age", ylab = "Frecuencia")
```



Se evidencia que hay pico de 40 a 45 años, seguidamente se observa que hay mas personas con edades entre 50 y 60 años.

Seguidamente se realiza un histograma para conocer la distribución de la variable thalachh.

```
hist(corazon$thalachh, main = "Histograma de thalachh", # Frecuencia
      xlab = "thalachh", ylab = "Frecuencia")
```



Se observa que la frecuencia mas comun de los valores maximos de presión arterial van 140 a 180.

2.4.2 Comprobación de la normalidad y homogeneidad de la varianza

Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la prueba de normalidad de **Anderson-Darling**.

Así, se comprueba que para que cada prueba se obtiene un p-valor superior al nivel de significación prefijado $\alpha = 0,05$. Si esto se cumple, entonces se considera que variable en cuestión sigue una distribución normal.

```
library(nortest)
alpha = 0.05
col.names = colnames(corazon)
for (i in 1:ncol(corazon)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n\n")
  if (is.integer(corazon[,i]) | is.numeric(corazon[,i])) {
    p_val = ad.test(corazon[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Formato de salida
      if (i < ncol(corazon) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal:
##
```



```
## age, sex, cp,
## trtbps, chol, fbs,
## restecg, thalachh, exng,
## oldpeak, slp, caa,
## thalloutput
```

También podemos para asumir normalidad de la muestra. observar **gráfico de cuantiles** y el **histograma** de cada una de las variables numéricas, además de aplicar el **test de Shapiro-Wilk**

Podemos comprobar que todas las variables pueden ser aproximadas a una distribución normal, a pesar de que no estén normalizadas. Como el tamaño de la muestra es superior a 30, podemos aplicar el **teorema del límite central** y asumir normalidad en la distribución de la muestra.

Estudiamos ahora la homogeneidad de varianzas mediante la aplicación de un **test de Fligner-Killeen**. En este caso, estudiaremos la homogeneidad en cuanto al colesterol los grupos conformados por mujeres frente a los hombres.

```
fligner.test(chol ~ sex, data = corazon)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: chol by sex
## Fligner-Killeen:med chi-squared = 9.0547, df = 1, p-value = 0.00262
```

No podemos concluir que existe homogeneidad en la varianza de los dos grupos, ya que el p-valor resultante del test es menor al nivel de significancia 0.05 marcado por el nivel de confianza.

Igualmente se realiza lo mismo para estudiar la homogeneidad de varianzas mediante la aplicación de un **test de Fligner-Killeen**, para la tensión arterial máxima frente a la edad.

```
fligner.test(age ~ sex, data = corazon)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: age by sex
## Fligner-Killeen:med chi-squared = 0.517, df = 1, p-value = 0.4721
```

Podemos concluir que existe homogeneidad en la varianza de los dos grupos, ya que el p-valor resultante del test es mayor al nivel de significancia 0.05 marcado por el nivel de confianza.

2.5. Pruebas estadísticas

2.5.1. ¿Qué variables cuantitativas influyen más en el diagnóstico?

En primer lugar, procedemos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre el diagnóstico final del paciente.

Para ello, se utilizará el coeficiente de correlación de Spearman, puesto que hemos visto que tenemos datos que no siguen una distribución normal.

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")

# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "output"

for (i in 1:(ncol(corazon.numeric) - 1)) {
  if (is.integer(corazon[,i]) | is.numeric(corazon[,i])) {
```

```

spearman_test = cor.test(corazon[,i],
                        corazon[,length(corazon)],
                        method = "spearman",
                        exact = FALSE)
corr_coef = spearman_test$estimate
p_val = spearman_test$p.value
# Add row to matrix
pair = matrix(ncol = 2, nrow = 1)
pair[1][1] = corr_coef
pair[2][1] = p_val
corr_matrix <- rbind(corr_matrix, pair)
rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(corazon)[i]
}
}

par(mfrow=c(1,1))
print(corr_matrix)

```

```

##           estimate      p-value
## age      -0.2384001 2.749629e-05
## sex      -0.2809366 6.678692e-07
## cp        0.4608602 2.444718e-17
## trtbps   -0.1215928 3.437373e-02
## chol     -0.1208882 3.543860e-02
## fbs      -0.1208882 3.543860e-02
## restecg   0.1486115 9.581603e-03
## thalachh  0.4283699 5.938298e-15
## exng     -0.4367571 1.520814e-15
## oldpeak  -0.4214871 1.766192e-14
## slp       0.3714605 2.393250e-11
## caa      -0.4576075 4.351795e-17
## thall    -0.4032993 2.799358e-13

```

Así, identificamos cuáles son las variables más correlacionadas con el diagnóstico en función de su proximidad con los valores -1 y +1. Teniendo esto en cuenta, queda patente cómo la variable más relevante en la fijación del diagnóstico es el cp : **Tipo de dolor en el pecho del paciente.**

Nota. Para cada coeficiente de correlación se muestra también su p-valor asociado, puesto que éste puede dar información acerca del peso estadístico de la correlación obtenida.

El valor más alto del coeficiente de correlación (0.460) que encontramos respecto a la variable “Output” - (Posee enfermedad Cardiovascular), es con la variable de la **máxima frecuencia cardíaca alcanzada**

2.5.2. ¿La frecuencia cardíaca del paciente es superior si su género es Hombre?

La segunda prueba estadística que se aplicará consistirá en un contraste de hipótesis sobre dos muestras para determinar si el diagnóstico es positivo dependiendo del tipo de Género del que se trate (femenino o masculino). Para ello, tendremos dos muestras: la primera de ellas se corresponderá a los diagnósticos de los pacientes mujeres, la segunda, con aquellos que pacientes hombres.

Se debe destacar que un test paramétrico como el que a continuación se utiliza necesita que los datos sean normales, si la muestra es de tamaño inferior a 30. Como en nuestro caso, $n > 30$, el contraste de hipótesis siguiente es válido (aunque podría utilizarse un test no paramétrico como el de Mann-Whitney, que podría resultar ser más eficiente para este caso).

```
corazon.mujeres.diag <-
corazon[corazon$sex == "0",]$output
corazon.hombre.diag <-
corazon[corazon$sex == "1",]$output
```

Así, se plantea el siguiente contraste de hipótesis de dos muestras sobre la diferencia de medias, el cual es unilateral atendiendo a la formulación de la hipótesis alternativa

$$H_0 : \mu_1 - \mu_2 = 0$$

$$H_1 : \mu_1 - \mu_2 > 0$$

donde μ_1 es la media de la población de la que se extrae la primera muestra y μ_2 es la media de la población de la que extrae la segunda. Así, tomaremos $\alpha = 0, 05$.

```
t.test(corazon.hombre.diag, corazon.mujeres.diag,
alternative = "less")

##
## Welch Two Sample t-test
##
## data: corazon.hombre.diag and corazon.mujeres.diag
## t = -5.3372, df = 209.95, p-value = 1.22e-07
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.2076342
## sample estimates:
## mean of x mean of y
## 0.4492754 0.7500000
```

Puesto que obtenemos un p-valor menor que el valor de significación fijado, rechazamos la hipótesis nula. Por tanto, podemos concluir que, efectivamente, que es mas probable que diagnostico una persona es positivo, si es mujer.

2.5.3. Modelo de regresión lineal

Tal y como se planteó en los objetivos de la actividad, resultará de mucho interés poder realizar predicciones sobre el diagnostico de enfermedad cardiovascular de una persona dadas sus características. Así, se calculará un modelo de regresión lineal utilizando regresores tanto cuantitativos como cualitativos con el que poder realizar las predicciones del diagnostico.

Para obtener un modelo de regresión lineal considerablemente eficiente, lo que haremos será obtener varios modelos de regresión utilizando las variables que estén más correladas con respecto al output, según la tabla obtenido en el apartado 2.1.5. Así, de entre todos los modelos que tengamos, escogeremos el mejor utilizando como criterio aquel que presente un mayor coeficiente de determinación (R^2).

```
# Regresores cuantitativos con mayor coeficiente
# de correlación con respecto al diagnostico
frecuenciaArterial = corazon$thalachh

# Regresores cualitativos
tipodolor = corazon$cp
electrocardiograficosReposo = corazon$restecg
STEjercicio = corazon$slp

# Variable a predecir
dianostico = corazon$output
```

```
# Generación de varios modelos
modelo1 <- lm(output ~ cp + restecg + thalachh +slp, data = corazon)
modelo2 <- lm(output ~ cp + restecg + thalachh +slp +chol, data = corazon)
```

Para los anteriores modelos de regresión lineal múltiple obtenidos, podemos utilizar el coeficiente de determinación para medir la bondad de los ajustes y quedarnos con aquel modelo que mejor coeficiente presente

```
# Tabla con los coeficientes de determinación de cada modelo
tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
2, summary(modelo2)$r.squared),
ncol = 2, byrow = TRUE)

colnames(tabla.coeficientes) <- c("Modelo", "R^2")
print(tabla.coeficientes)
```

```
##      Modelo      R^2
## [1,]      1 0.3295705
## [2,]      2 0.3314125
```

En este caso, tenemos que el cuarto modelo es el más conveniente dado que tiene un mayor coeficiente de determinación. Ahora, empleando este modelo, podemos proceder a realizar predicciones de diagnóstico de personas como la siguiente

```
newdata <- data.frame(
cp = 3,
restecg = 0,
thalachh = 150,
slp = 0,
chol = 223)

# Predecir el precio
predict(modelo2, newdata)
```

```
##      1
## 0.6058435
```

2.6.Conclusiones

Como se ha visto, se han realizado diferentes tipos de pruebas estadísticas sobre un conjunto de datos que se correspondía con diferentes variables relativas al diagnóstico de personas a sufrir o no enfermedades cardíacas, con motivo de cumplir con lo planteado en la actividad. Para cada una de ellas, hemos podido ver cuáles son los resultados que arrojan (entre otros, mediante tablas) y qué conocimientos pueden extraerse a partir de ellas.

Así, el análisis de correlación y el contraste de hipótesis nos ha permitido conocer cuáles de estas variables ejercen una mayor influencia sobre el precio final del coche, mientras que el modelo de regresión lineal obtenido resulta de utilidad a la hora de realizar predicciones para esta variable dadas unas características concretas.

Previamente, se han sometido los datos a un preprocesamiento para manejar los casos de ceros o elementos vacíos y valores extremos (outliers). Para el caso del primero, se ha hecho uso de un método de imputación de valores de tal forma que no tengamos que eliminar registros del conjunto de datos inicial y que la ausencia de valores no implique llegar a resultados poco certeros en los análisis. Para el caso del segundo, el cual constituye un punto delicado a tratar, se ha optado por incluir los valores extremos en los análisis dado que parecen no resultar del todo atípicos si los comparamos con los valores que toman las correspondientes variables para coches que existen en el mercado actual.

Se logro determinar que el modelo tiene un buen resultado ya que se realizo pruebas con los intriduciendo los datos del data set y la predicción concidio con la dariable de salida del data set.

3.Referencia

- Calvo M, Subirats L, Pérez D (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
- Tutorial de Github (<https://guides.github.com/activities/hello-world/>)
- Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
- Wes McKinney (2012). Python for Data Analysis. O Reilly Media, Inc.