Tipología y ciclo de vida : PRA2 - Selección y preparación de un juego de datos

Autores: Óscar López Montero y Jose Antonio Jara

Abril 2021

Contents

---

1 Introducción

1.1 Presentación

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.

Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github (https://github.com) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo.

Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github.Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía:

1.2 Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.

Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

1.3 Objetivos

Los objetivos concretos de esta práctica son:

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.

1.4 Descripción de la Práctica a realizar

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle (https://www.kaggle.com).

Algunos ejemplos de dataset con los que podéis trabajar son:

Red Wine Quality (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009)

Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic)

El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en esta competición.

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?
2. Integración y selección de los datos de interés a analizar.

3. Limpieza de los datos.

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

   3.2. Identificación y tratamiento de valores extremos.

4. Análisis de los datos.

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

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.

1.5 Recursos Básicos

Los siguientes recursos son de utilidad para la realización de la práctica:

Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.

Megan Squire (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'Reilley Media, Inc.

Tutorial de Github https://guides.github.com/activities/hello-world

1.6 Criterios de valoración

Todos los apartados son obligatorios. La ponderación de los ejercicios es la siguiente:

Los apartados 1, 2 y 6 valen 0,5 puntos.

Los apartados 3, 5 y 7 valen 2 puntos.

El apartado 4 vale 2,5 puntos.

Se valorará la idoneidad de las respuestas, que deberán ser claras y completas. Las diferentes etapas deberán justificarse y acompañarse del código correspondiente. También se valorará la síntesis y claridad, a través del uso de comentarios, del código resultante, así como la calidad de los datos finales analizados.

Ejercicios prácticos

Para todas las PRA es **necesario documentar** en cada apartado del ejercicio práctico que se ha hecho y como se ha hecho.

1.7 Formato y fecha de entrega PRA_1

Durante la semana del 24 al 28 de mayo el grupo podrá entregar al profesor una entrega parcial opcional. Esta entrega parcial es muy recomendable para recibir asesoramiento sobre la práctica y verificar que la dirección tomada es la correcta. Se entregarán comentarios a los estudiantes que hayan efectuado la entrega parcial pero no contará para la nota de la práctica. En la entrega parcial los estudiantes deberán entregar por correo electrónico, al profesor encargado del aula, el enlace al repositorio Github con el que hayan avanzado.

En referente a la entrega final, hay que entregar un único fichero que contenga el enlace Github, el cual no se podrá modificar posteriormente a la fecha de entrega, donde haya:

1. Una Wiki con los nombres de los componentes del grupo y una descripción de los ficheros.
2. Un documento PDF con las respuestas a las preguntas y los nombres de los componentes del grupo. Además, al final del documento, deberá aparecer la siguiente tabla de contribuciones al trabajo, la cual debe firmar cada integrante del grupo con sus iniciales. Las iniciales representan la confirmación de que el integrante ha participado en dicho apartado. Todos los integrantes deben participar en cada apartado, por lo que, idealmente, los apartados deberían estar firmados por todos los integrantes..
3. Una carpeta con el código generado para analizar los datos.
4. El fichero CSV con los datos originales.
5. El fichero CSV con los datos finales analizados. Este documento de entrega final de la Práctica 2 se debe entregar en el espacio de Entrega y Registro de AC del aula antes de las 23:59 del día 8 de junio. No se aceptarán entregas fuera de plazo.

1.8 Nota: Propiedad intelectual

A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por lo tanto comprensible hacerlo en el marco de una práctica de los estudios de Informática, Multimedia y Telecomunicación de la UOC, siempre y cuando esto se documente claramente y no suponga plagio en la práctica.

Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se debe presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra esta protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL …). El estudiante deberá asegurarse de que la licencia no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente tendrá que asumir que la obra esta protegida por copyright.

Deberéis, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.

---

2 Resolución de la práctica

En esta práctica, hemos decidido trabajar sobre el dataset Heart Attack Analysis & Prediction Dataset situado en el repositorio de Kaggle.com. El dataset contiene diferentes muestras sobre pacientes susceptibles a recibir ataques al corazón.

2.1 Descripción del Dataset

El dataset sobre el que vamos a realizar el análisis y predicción trata sobre enfermedades cardiovasculares, en concreto, de la susceptibilidad de algunas personas a recibir ataques al corazón.

El objetivo de la práctica es analizar un conjunto de datos con el fin de identificar que tipo de factores son los más influyentes sobre los pacientes que han recibido ataques al corazón, y con ello, poder predecir con mayor facilidad cuando es más probable que ocurra y qué tipo de personas sufren de un mayor riesgo.

Actualmente en España existen más de 10 millones de personas con enfermedades y/o patologías relacionadas con el corazón, y más de 120.000 fallecen como consecuencia de estas. Entre todas estás personas, sólo en España, más de 14.000 personas fallecieron anualmente a causa de un infarto agudo de miocardio.

La predicción en temas de salud es de gran ayuda a la hora de tratar a algunos pacientes con el fin de que el diagnóstico de su médico sea lo más preciso posible, poder predecir qué personas están más expuestas a este tipo de problemas cardiovasculares, así como los principales causantes puede ayudar a más de un médico a la hora de recomendar un tratamiento a su cliente, llegando a salvar más de una vida.

Antes de explicar las variables que vamos a utilizar en nuestro dataset, y que vamos a analizar con el fin de determinar su incidencia en los infartos agudos de miocardio, vamos a explicar de qué tratan, así como algunas de las pruebas que los médicos realizan a sus pacientes, con el fin de entender los procedimientos y estudios que son realizados hoy en día.

Un infarto agudo de miocardio o ataque al corazón es provocado por la muerte de células cardiacas debido al desequilibrio entre el aporte de riego sanguineo por la circulación coronaria y la demanda del mismo. Este suministro deficiente de oxígeno al corazón es el resultante de anginas de pecho, las cuales, suelen preceder a los infartos si no es tratada, ya que acaban produciendo la muerte celular explicada anteriormente.

Las variables que componen nuestro dataset y que vamos a utilizar para estudiar estos casos son:

- age : Edad del paciente.
- sex : Género del paciente.
- cp : Tipo de dolor en el pecho
    0. Asintomático.
    1. Angina típica.
    2. Angina atípica.
    3. Dolor no-anginal.
- trtbps : Presión arterial en reposo (en mm Hg)
- chol : Colesterol (en mg/dl)
- fbs : (Glucemia > 120 mg/dl) (1 = True; 0 = False)
- restecg : Resultados de electrocardiograma en reposo.
    0. Muestra una probable hipertrófia del ventrículo izquierdo.
    1. Anomalías de onda ST-T (Inversiones de onda T y/o una elevación de la onda en el segmento ST > 0.05 mV)
    2. Normal
- thalachh : Ritmo cardiaco máximo.
- exng: Si el paciente ha sufrido una angina provocada por el ejercicio (1 = Si, 0 = No)
- oldpeak: Depresión en el segmento ST al hacer ejercicio en relación con el reposo.
- slp: Pendiente del segmento ST en la electrocardiograma.
    0. Pendiente descendente.
    1. Plano.
    2. Pendiente ascendente.

- caa: Número de vasos principales del corazón (De 0 a 3).
- thall: Prueba de esfuerzo cardiaco (Thallium stress test).
    1. El test muestra un defecto irreversible.
    2. El flujo sanguineo se encuentra dentro de los valores normales.
    3. El test muestra un defecto reversible.
- target : Variable de clase. 0 equivale a un menor riesgo de ataque cardiaco mientras que 1 equivale un mayor riesgo.

Tras las variables, vamos a explicar de una manera más informada el significado de las mismas.

En cada latido del corazón, el impulso cardiaco sucede en la nódulo sinusal (Situado en la aurícula derecha), para después diseminarse por las aurículas, produciendo la despolarización y por consiguiente, la contracción de las mismas. Tras esto, llega al nódulo aurioventricular, situada en la parte izquierda de la aurícula derecha, donde la onda eléctrica sufre una breve pausa de 100ms. Tras esta pausa, la onda eléctrica se disemina a través del *haz de His* (Un fino cordón muscular) llegando a los ventrículos y despolarizándolos y ocasionando la contracción ventricular.

Un electrocardiograma es una representación visual de la actividad eléctrica del corazón en función del tiempo. Este, consta principalmente de 5 fases:

1. Onda P: La primera ligera curva hacia arriba que aparece en el electrocardiograma (ECG). Es el momento en el que las aurículas se contraen y envian sangre hacia los ventrículos.
2. Segmento P-R: Periodo entre la Onda P y la siguiente deflexión o curva, en este las aurículas están terminando de vaciarse.
3. Complejo QRS: Periodo en el que los ventrículos se cotraen, expulsando su contenido. Este complejo está compuesto por las ondas Q,R y S. La onda Q no siempre aparece en el ECG, pero se caracteriza por ser la primera pequeña deflexión negativa de este complejo. Esta es seguida por la onda R, que varia en altura dependiendo de las condiciones físicas del paciente (A mayores capacidades físicas, mayor altura). La onda S es la continuación de la onda R, y se caracteriza por ser la fase decreciente del complejo.
4. Segmento ST: Es la fase que más vamos a estudiar en este caso. Se trata del trazado lineal entre la onda S y la onda T. Su pendiente en relación con la linea basal puede llegar a significar insuficiencia de riego cardiaco. Elevaciones y depresiones en este segmento superiores a 1 mm pueden llegar a significar la oclusión de una arteria coronaria.
5. Onda T: También utilizada en nuestras variables. La onda T consiste en una pequeña deflexión positiva que refleja la repolarización ventricular o el momento en el que el corazón se encuentra en estado de relajación tras expulsar la sangre que se hallaba en los ventrículos. En los casos en los que esta onda está invertida, si la inversión es superior a 1mm suele asociarse a casos de isquemia miocárdica.

Por otro lado, vamos a explicar también de qué trata el Thallium stress test o prueba de esfuerzo cardiaco.

Esta prueba es realizada con el fin de mostrar cómo fluye la sangre en el corazón mientras se hace ejercicio o se está en reposo, para ver la capacidad del corazón para responder al estrés físico en un entorno clínico controlado comparando la circulación coronaria en reposo con la obtenida mientras se está en el momento de esfuerzo cardiaco cumbre.

Para realizar el test, se introduce un líquido con una pequeña cantidad de radioactividad llamado radioisótopo en el flujo sanguineo, este fluirá por tu sangre hasta llegar al corazón para más tarde poder ser inspeccionado por una cámara gamma con el fin de detectar discrepancias en el funcionamiento del músculo.

2.2 Integración y selección de los datos de interés

Para esta práctica vamos a utilizar todos los datos que nos otorga el dataset, pese a tener distinta influencia a la hora de que un paciente sea considerado de riesgo.

Leemos y revisamos los datos:

```
heartdata<-read.csv('../data/heart.csv')
summary(heartdata)
```

```
##       age             sex               cp            trtbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0
##       chol            fbs            restecg          thalachh
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :240.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.3   Mean   :0.1485   Mean   :0.5281   Mean   :149.6
##  3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##       exng            oldpeak          slp             caa
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000
##      thall           output
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:2.000   1st Qu.:0.0000
##  Median :2.000   Median :1.0000
##  Mean   :2.314   Mean   :0.5446
##  3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000
```

```
str(heartdata)
```

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

Podemos ver como las variables categóricas aún están representadas por valores numéricos, vamos a trans-

formarlas con el fin de hacer el análisis más visual y entendible.

```
heartdata$output <- as.factor(heartdata$output)
levels(heartdata$output) <- c("Bajo riesgo", "Alto riesgo")
summary(heartdata$output)
```

```
## Bajo riesgo Alto riesgo
##         138         165
```

```
heartdata$sex <- as.factor(heartdata$sex)
levels(heartdata$sex) <- c("Mujer", "Hombre")
summary(heartdata$cp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   0.967   2.000   3.000
```

```
heartdata$cp <- as.factor(heartdata$cp)
levels(heartdata$cp) <- c("Asintomático","Angina típica","Angina atípica",
                          "Dolor no anginal")
heartdata$fbs <- ifelse(heartdata$fbs ==1, T,F)
#heartdata$fbs <- as.logical(heartdata$fbs)
heartdata$fbs <- as.factor(heartdata$fbs)
summary(heartdata$fbs)
```

```
## FALSE  TRUE
##   258    45
```

```
heartdata$restecg <- as.factor(heartdata$restecg)
levels(heartdata$restecg) <- c("Normal","Anomalía de onda ST-T","Hipertrofia")
heartdata$exng <- ifelse(heartdata$exng ==1, T,F)
#heartdata$exng <- as.logical(heartdata$exng)
heartdata$exng <- as.factor(heartdata$exng)
heartdata$slp <- as.factor(heartdata$slp)
levels(heartdata$slp) <- c("Pendiente ascendente","Plano",
                           "Pendiente descendente")
heartdata$thall <- as.factor(heartdata$thall)
summary(heartdata$thall)
```

```
##   0   1   2   3
##   2  18 166 117
```

En el caso de la prueba de esfuerzo cardiaco, tenemos variables nulas mapeadas a 0, dado que no tenemos este dato, que son únicamente dos registros y que no podemos estimarlos mediante modelos lineales, vamos a eliminar estos dos casos.

```
heartdata <- heartdata[heartdata$thall!=0,]
heartdata$thall<-droplevels(heartdata$thall)
levels(heartdata$thall) <- c("Defecto irreversible","Normal",
                             "Defecto reversible")
summary(heartdata$thall)
```

```
## Defecto irreversible              Normal   Defecto reversible
##                   18                 166                 117
```

2.3 Limpieza de los datos

2.3.1 Tratamiento de elementos vacíos

Procedemos a comprobar y eliminar/reemplazar los valores nulos de nuestro dataset. Aunque no parecen contener campos vacíos tras la primera toma de contacto y haberlos revisado previamente en Kaggle.

```
colSums(is.na(heartdata))
```

```
##      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
```

Vemos como no tenemos ningún valor vacío en nuestro conjunto de datos. En el caso de encontrarnos un valor vacío en una variable categórica, y que esta sea importante para el análisis se eliminaría. Por otro lado, si tenemos un valor vacío en una variable cuantitativa, podriamos estimarlo a traves de un modelo de regresión junto a otra variable cuantitativa con la que mantenga buena correlación.

Tras comprobarlo, guardamos los datos tras ser tratados en un nuevo fichero.

```
write.csv(heartdata, "../data/heart_clean.csv")
```

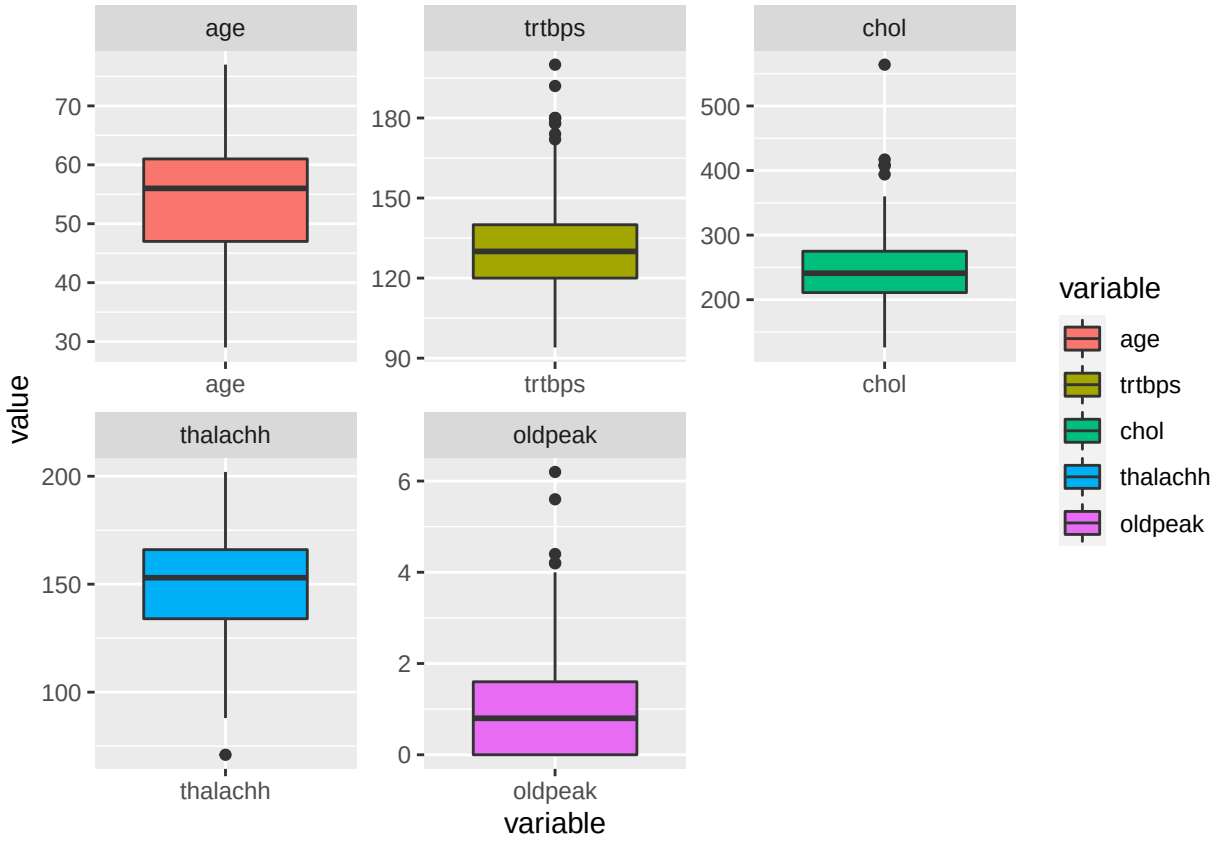
2.3.2 Identificación y tratamiento de valores extremos

Vamos a realizar diagramas de caja y bigotes sobre las variables cuantitativas con el fin de encontrar outliers o valores atípicos.

Unimos los datos cuantitativos usando la función melt de la librería reshape.

Y mostramos los diagramas usando ggplot.

```
library(reshape2)
library(ggplot2)
hd.m <- melt(heartdata[c(1,4,5,8,10)], id.var = NULL)
gbp <-ggplot(hd.m,aes(x=variable,y=value, fill=variable)) + geom_boxplot()
gbp + facet_wrap( ~ variable, scales="free")
```



Podemos observar tal vez un outlier que destaca en la variable que mide la cantidad del colesterol en sangre. Investigamos su valor, así como si se puede tratar de un caso real.

```
summary(heartdata$chol)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   126.0   211.0   241.0   246.5   275.0   564.0
```

El nivel saludable de colesterol se encuentra entre 125 y 200 mg/dL. Existen casos en los que se dan niveles elevados de colesterol debidos a trastornos genéticos de origen familiar, ergo no vamos a eliminar el registro.

Por otro lado, también existen algunos outliers que nos llaman la atención respecto a la presión arterial en reposo de los pacientes, comprobamos el valor máximo de la variable.

```
summary(heartdata$trtbps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    94.0   120.0   130.0   131.6   140.0   200.0
```

Vemos que llega hasta 200. Se pueden dar casos de personas que tienen esta presión arterial en reposo cuando están sufriendo una crisis hipertensiva, usualmente acompañada de un dolor agudo en el pecho.

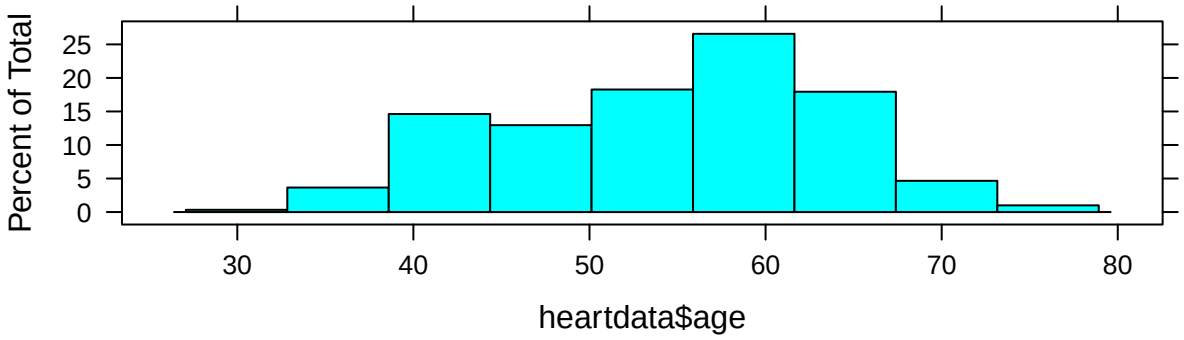
Una vez analizados los datos, no vemos coherente modificar o eliminar ninguno de estos valores, por tanto, procedemos a analizarlos junto al resto de registros del dataset.

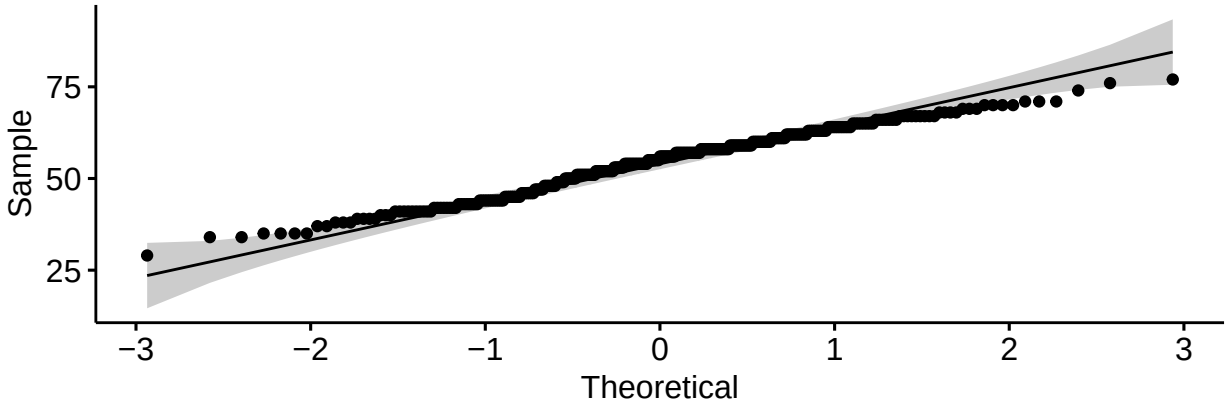
2.4 Análisis de los datos

2.4.1 Comprobación de la normalidad y homogeneidad de la varianza

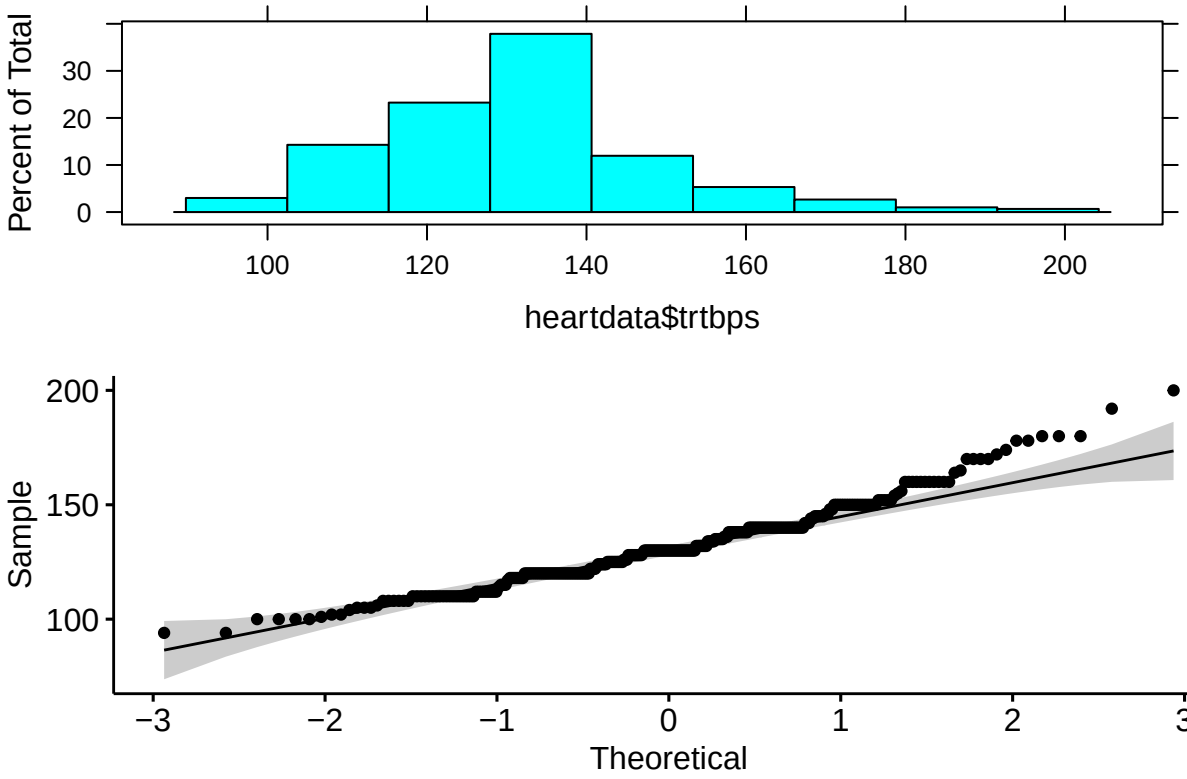
Estudio de la normalidad en las variables numéricas

```
library(ggpubr)
library(ggplot2)
library(lattice)
library(grid)
library(gridExtra)

# Varible Age
H1 <- histogram(heartdata$age)
G1 <- ggqqplot(heartdata$age)

# Variable trtbps
H2 <- histogram(heartdata$trtbps)
G2 <- ggqqplot(heartdata$trtbps)

# Variable chol
H3 <- histogram(heartdata$chol)
G3 <- ggqqplot(heartdata$chol)

# Variable thalach
H4 <- histogram(heartdata$thalach)
G4 <- ggqqplot(heartdata$thalach)

# Variable oldpeak
H5 <- histogram(heartdata$oldpeak)
G5 <- ggqqplot(heartdata$oldpeak)

# Mostramos las gráficas
grid.arrange(H1, G1)
```

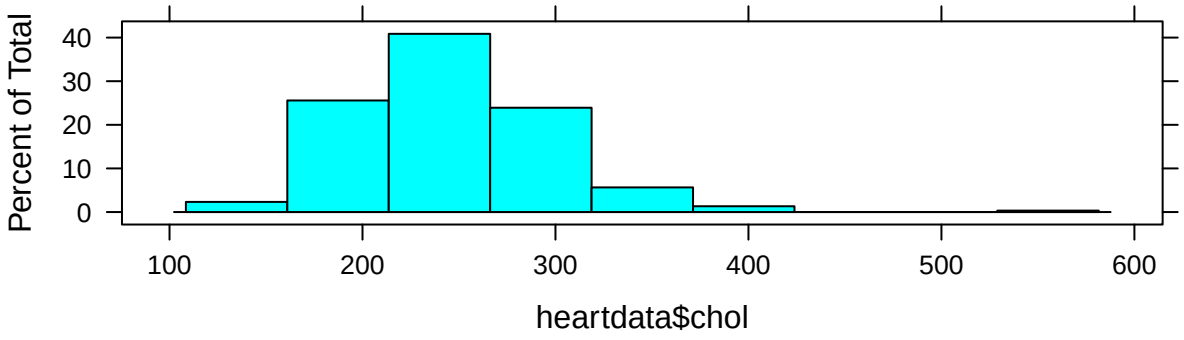


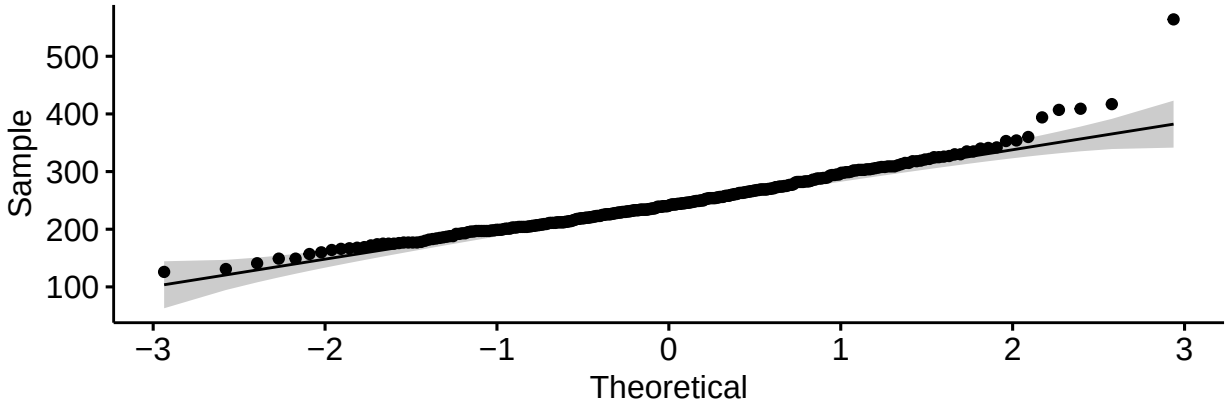



```
grid.arrange(H2, G2)
```

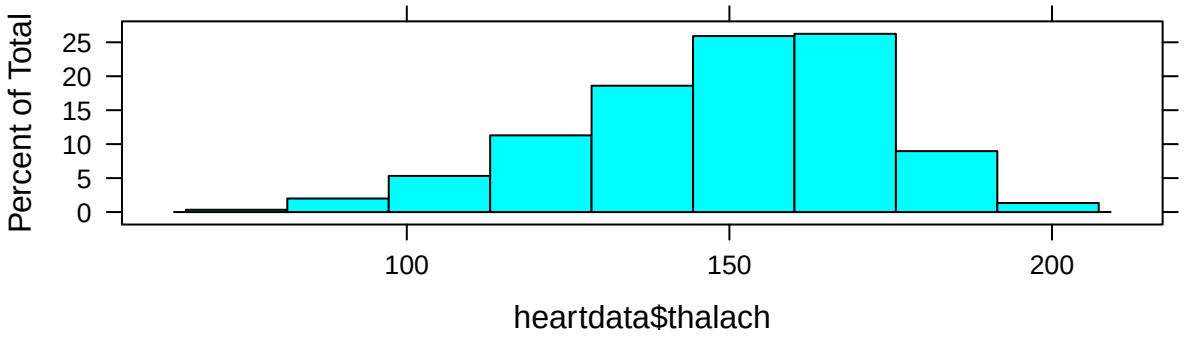


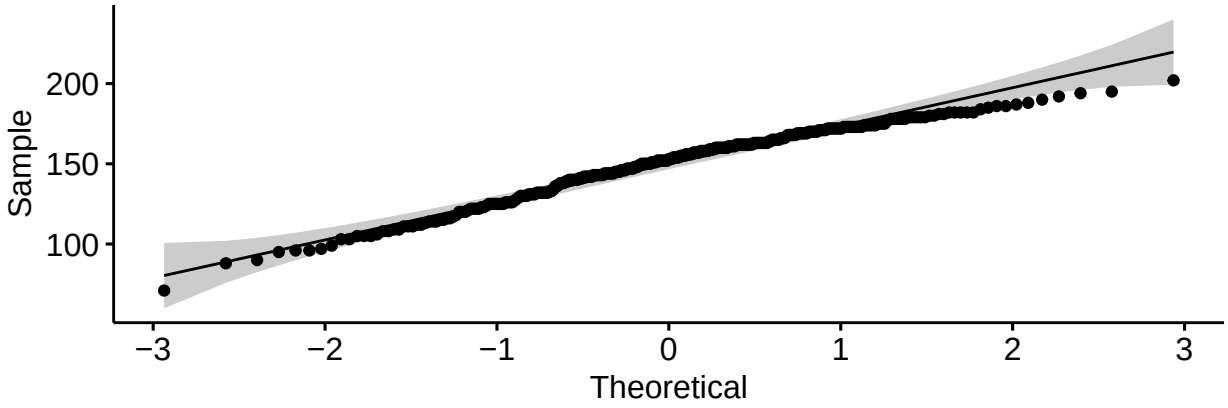

```
grid.arrange(H3, G3)
```



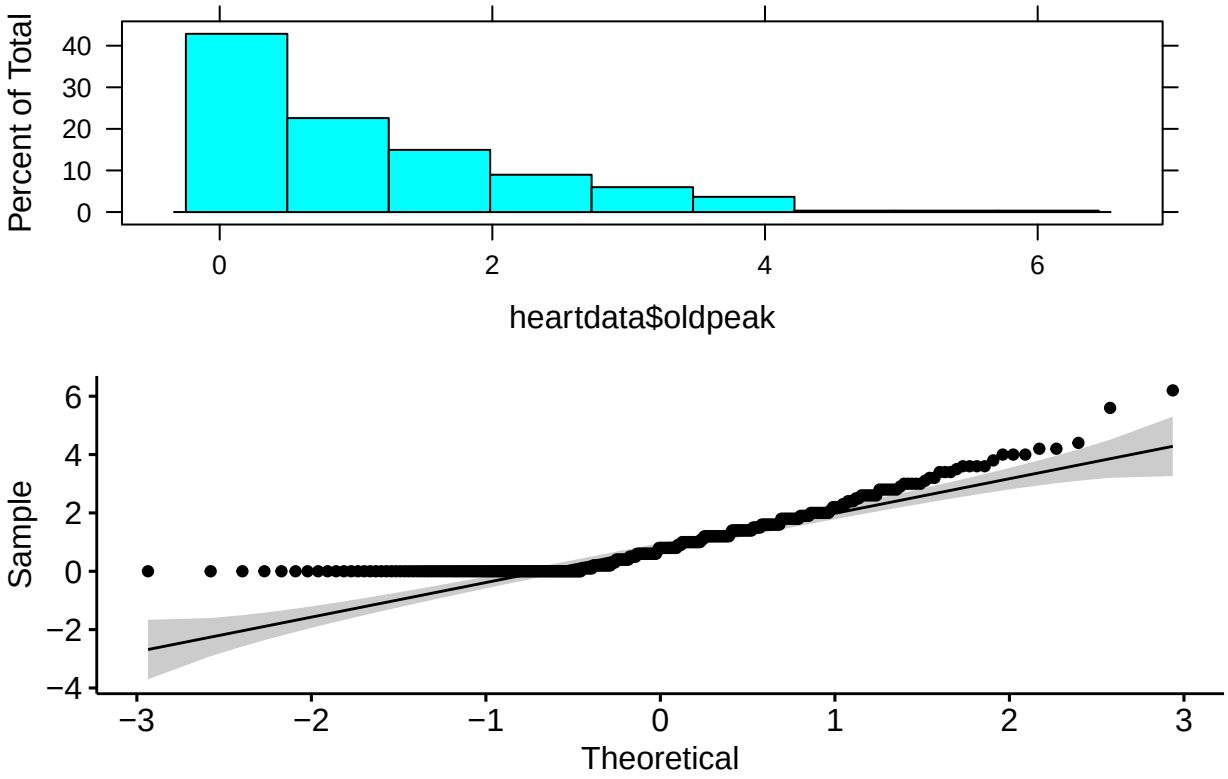



```
grid.arrange(H4, G4)
```






```
grid.arrange(H5, G5)
```



Los gráficos de densidad y Q-Q nos confirman que todas las variables numéricas siguen una distribución normal. Practicamente todas las gráficas tienen una normalidad en el centro de la muestra y están un poco desviadas en los extremos, pero este dato no es suficiente para descartar la normalidad, además todas las variables tienen n>30 porque podemos asumir que siguen una distribución normal.

Estudio de la homogeneidad de las variables numéricas.

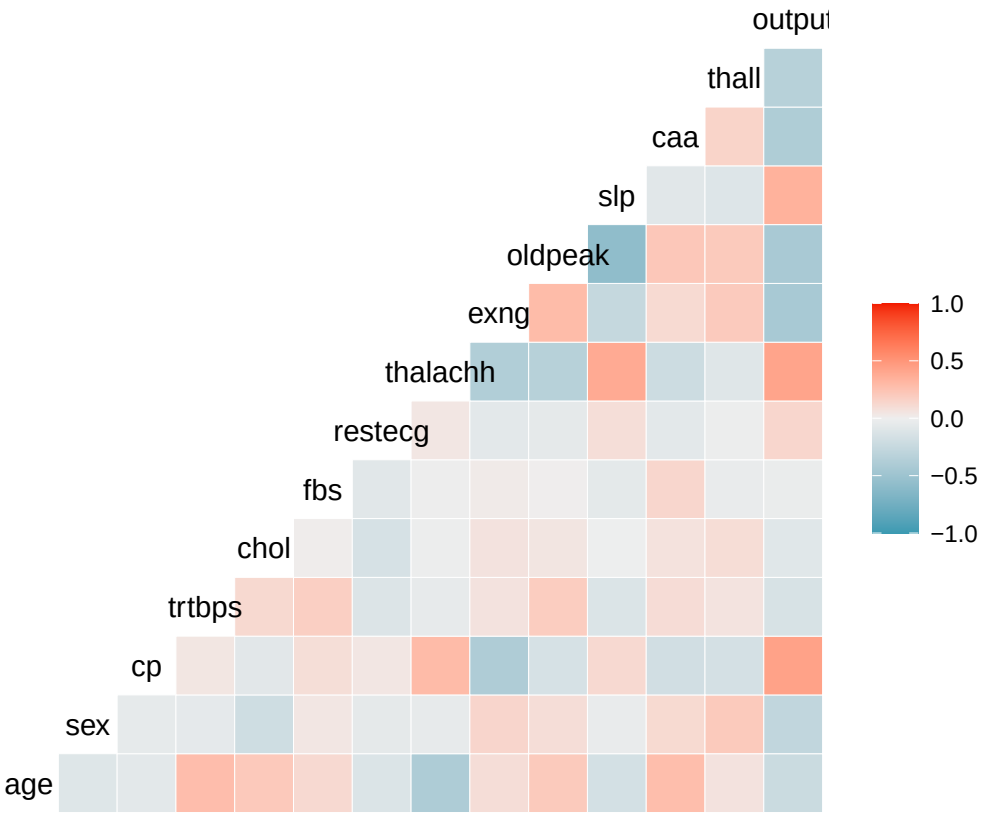
2.4.2 Pruebas estadísticas y métodos de análisis

Para realizar este estudio aplicaremos 4 modelos: Matriz de correlación, regresión lineal, árbol de decisión y agrupación(clustering).

2.4.2.1 Matriz correlación Mediante la matriz de correlación podremos observar que nivel de correlación guardan la variables entre ellas y de esta manera saber cuales son la más significativas.

Utilizaremos el coeficiente de correlación de Pearson para ver cuán asociadas se encuentra la variables.

```
library(GGally)
#Utilizaremos el data set original ya que todos los campos son numéricos.
heart_ori <- read.csv('../data/heart.csv')

ggcorr(heart_ori, method = c("everything", "pearson"))
```




```
cor(heart_ori)
```

```
##                  age         sex          cp      trtbps         chol
## age       1.00000000 -0.09844660 -0.06865302  0.27935091  0.213677957
## sex      -0.09844660  1.00000000 -0.04935288 -0.05676882 -0.197912174
## cp       -0.06865302 -0.04935288  1.00000000  0.04760776 -0.076904391
## trtbps    0.27935091 -0.05676882  0.04760776  1.00000000  0.123174207
## chol      0.21367796 -0.19791217 -0.07690439  0.12317421  1.000000000
## fbs       0.12130765  0.04503179  0.09444403  0.17753054  0.013293602
## restecg  -0.11621090 -0.05819627  0.04442059 -0.11410279 -0.151040078
## thalachh -0.39852194 -0.04401991  0.29576212 -0.04669773 -0.009939839
## exng      0.09680083  0.14166381 -0.39428027  0.06761612  0.067022783
## oldpeak   0.21001257  0.09609288 -0.14923016  0.19321647  0.053951920
## slp      -0.16881424 -0.03071057  0.11971659 -0.12147458 -0.004037770
## caa       0.27632624  0.11826141 -0.18105303  0.10138899  0.070510925
## thall     0.06800138  0.21004110 -0.16173557  0.06220989  0.098802993
## output   -0.22543872 -0.28093658  0.43379826 -0.14493113 -0.085239105
##                 fbs     restecg     thalachh        exng      oldpeak
## age       0.121307648 -0.11621090 -0.398521938  0.09680083  0.210012567
## sex       0.045031789 -0.05819627 -0.044019908  0.14166381  0.096092877
## cp        0.094444035  0.04442059  0.295762125 -0.39428027 -0.149230158
## trtbps    0.177530542 -0.11410279 -0.046697728  0.06761612  0.193216472
## chol      0.013293602 -0.15104008 -0.009939839  0.06702278  0.053951920
## fbs       1.000000000 -0.08418905 -0.008567107  0.02566515  0.005747223
## restecg  -0.084189054  1.00000000  0.044123444 -0.07073286 -0.058770226
## thalachh -0.008567107  0.04412344  1.000000000 -0.37881209 -0.344186948
```

```
## exng      0.025665147 -0.07073286 -0.378812094  1.00000000  0.288222808
## oldpeak   0.005747223 -0.05877023 -0.344186948  0.28822281  1.000000000
## slp      -0.059894178  0.09304482  0.386784410 -0.25774837 -0.577536817
## caa       0.137979327 -0.07204243 -0.213176928  0.11573938  0.222682322
## thall    -0.032019339 -0.01198140 -0.096439132  0.20675379  0.210244126
## output   -0.028045760  0.13722950  0.421740934 -0.43675708 -0.430696002
##                  slp         caa       thall      output
## age      -0.16881424  0.27632624  0.06800138 -0.22543872
## sex      -0.03071057  0.11826141  0.21004110 -0.28093658
## cp        0.11971659 -0.18105303 -0.16173557  0.43379826
## trtbps   -0.12147458  0.10138899  0.06220989 -0.14493113
## chol     -0.00403777  0.07051093  0.09880299 -0.08523911
## fbs      -0.05989418  0.13797933 -0.03201934 -0.02804576
## restecg   0.09304482 -0.07204243 -0.01198140  0.13722950
## thalachh  0.38678441 -0.21317693 -0.09643913  0.42174093
## exng     -0.25774837  0.11573938  0.20675379 -0.43675708
## oldpeak  -0.57753682  0.22268232  0.21024413 -0.43069600
## slp       1.00000000 -0.08015521 -0.10476379  0.34587708
## caa      -0.08015521  1.00000000  0.15183213 -0.39172399
## thall    -0.10476379  0.15183213  1.00000000 -0.34402927
## output    0.34587708 -0.39172399 -0.34402927  1.00000000
```

En general podemos decir que no existe muchas relación entre las variables del dataset.

Nos centraremos en la relación entre la variable 'tarjet' (output) y el resto:

- Encontramos que slp, thalachh, restecg y cp son la que más peso tienen respecto a 'tarjet'(output) es decir, la pendiente del segmento ST en la electrocardiograma, ritmo cardíaco máximo, resultados de electrocardiograma en reposo y el tipo de dolor en el pecho.

Todas son pruebas médicas para evaluar el estado del corazón, por lo que es lógico que tengan un peso importante, porque a través de los resultados los médicos tienen datos fiables para el diagnóstico.

2.4.2.2 Regresión lineal

Mediante la regresión lineal podremos predecir si un paciente padecerá enfermedad cardíaca.

Vamos a utilizar las variables de mayor correlación obtenidas en el apartado anterior.

```r
# Aplicamos el algoritmo de regresión lineal
ml.heart <- lm(output~age+sex+slp+thalachh+restecg+cp, data = heart_ori,
               family = binomial)

# Resultados
summary(ml.heart)
```

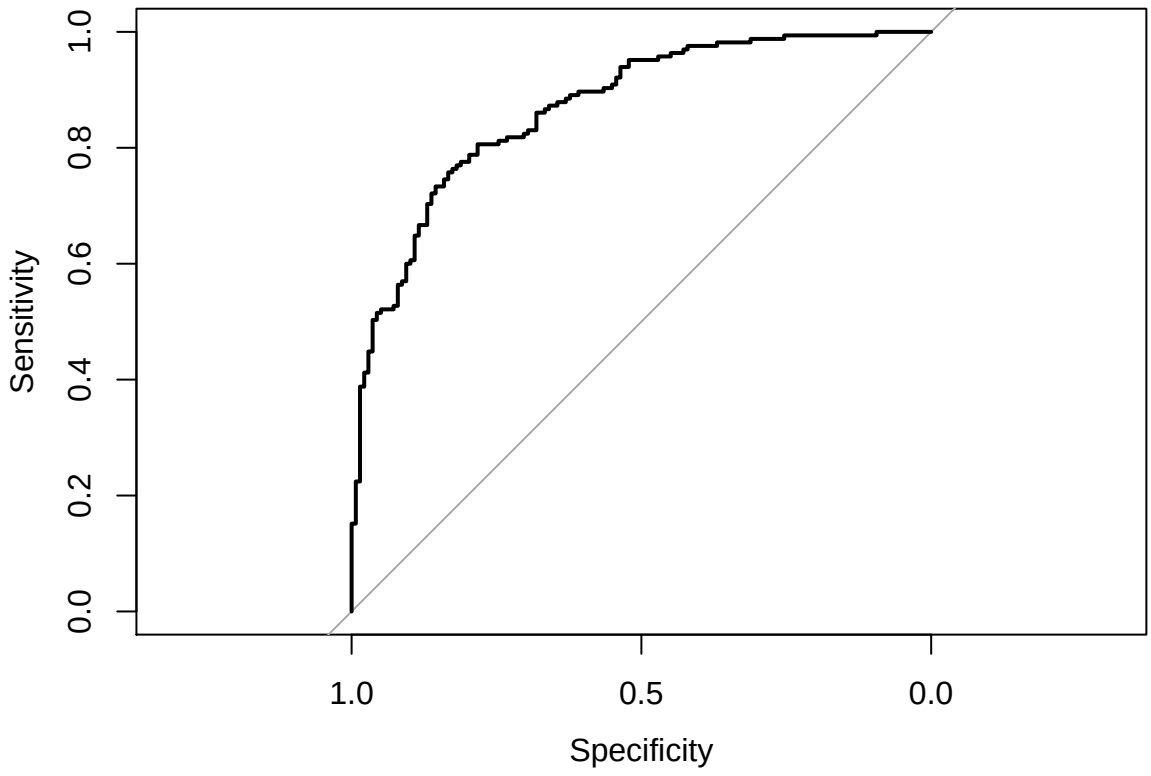
```
##
## Call:
## lm(formula = output ~ age + sex + slp + thalachh + restecg +
##     cp, data = heart_ori, family = binomial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98031 -0.26046 -0.00023  0.28813  0.97328
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.039111   0.272064   0.144 0.885791
```

```
## age         -0.006130   0.002744  -2.234 0.026230 *
## sex         -0.275356   0.048828  -5.639 3.99e-08 ***
## slp          0.162487   0.039702   4.093 5.51e-05 ***
## thalachh     0.004082   0.001193   3.420 0.000713 ***
## restecg      0.064165   0.043322   1.481 0.139633
## cp           0.159972   0.022867   6.996 1.76e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3907 on 296 degrees of freedom
## Multiple R-squared:  0.3988, Adjusted R-squared:  0.3866
## F-statistic: 32.72 on 6 and 296 DF,  p-value: < 2.2e-16
```

Podemos observar que la variables restecg no es significativa para el modelo de regresión ya que p-values > 0.05.

Para evaluar la eficiencia del modelo de regresión lineal vamos a utilizar la curva ROC.

```
library(pROC)
prob = predict(ml.heart, heart_ori, type="response")
r = roc(heart_ori$output , prob, data=heart_ori)
plot(r)
```




```
r
```

```
##
## Call:
## roc.default(response = heart_ori$output, predictor = prob, data = heart_ori)
##
```

```
## Data: prob in 138 controls (heart_ori$output 0) < 165 cases (heart_ori$output 1).
## Area under the curve: 0.8697
```

El resultado del modelo es de 0.8697, por lo que la habilidad del modelo para predecir enfermedades cardíacas es muy bueno.

AURO > 0.8, el modelo discrimina de modo muy bueno.

Vamos usar el modelo mencionado para realizar una predicción.

```
# creamos un nuevo dataset con los datos del paciente

paciente <- data.frame(age=57,sex=1,slp=2,thalachh=162,restecg=2, cp=2)

# Algoritmo de predicción
predict(ml.heart, paciente)
```

```
##         1
## 0.8488944
```

Paciente con las siguientes patologías:

- Edad = 57
- sexo = Hombre
- Pendiente del segmento ST en la electrocardiograma = 2 - Pendiente ascendente
- Ritmo cardíaco máximo = 162 pulsaciones por minuto.
- Resultados de electrocardiograma en reposo = 2 - Normal
- Tipo de dolor en el pecho = 2 - Dolot no-anginal.

Tiene una probabilidad de un 85% de padecer una enfermedad cardíaca según nuestro modelo de regresión lineal con una efectividad del 83%.

2.4.2.3 Árbol de decisión con algoritmo C50

Desordenamos la muestra mediante un algoritmo de desordenación.

```
library(C50)
set.seed(10)
heart_r <- heartdata[sample(nrow(heartdata)),]
```

Separaremos los datos en train con una proporción 2/3 y test con una proporción de 1/3, esto es necesario para poder aplicar el algoritmo.

Clasificaremos los datos por la variable output que se encuentra en la posición 14 del dataset.

```
set.seed(666)
y <- heart_r[,14]
X <- heart_r[,1:13]

# Creamos rangos para los subconjuntos train (2/3) y test(1/3)

indexes = sample(1:nrow(heartdata), size=floor((2/3)*nrow(heartdata)))
trainX<-X[indexes,]
trainy<-y[indexes]
testX<-X[-indexes,]
testy<-y[-indexes]
```

Aplicamos el algoritmo C50.

```r
trainy = as.factor(trainy)
mo <- C50::C5.0(trainX, trainy,rules=TRUE )
summary(mo)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Jun 07 22:29:00 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 200 cases (14 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (34, lift 2.1)
##  cp in {Asintomático, Angina típica}
##  trtbps > 108
##  chol > 242
##  thall in {Defecto irreversible, Defecto reversible}
##  ->  class Bajo riesgo  [0.972]
##
## Rule 2: (50/1, lift 2.1)
##  cp in {Asintomático, Angina típica}
##  oldpeak > 0.4
##  thall in {Defecto irreversible, Defecto reversible}
##  ->  class Bajo riesgo  [0.962]
##
## Rule 3: (49/2, lift 2.0)
##  trtbps > 108
##  caa > 0
##  thall in {Defecto irreversible, Defecto reversible}
##  ->  class Bajo riesgo  [0.941]
##
## Rule 4: (45/4, lift 1.9)
##  cp in {Asintomático, Dolor no anginal}
##  trtbps <= 144
##  caa > 0
##  ->  class Bajo riesgo  [0.894]
##
## Rule 5: (26/1, lift 1.7)
##  chol <= 242
##  oldpeak <= 0.4
##  caa <= 0
##  ->  class Alto riesgo  [0.929]
##
## Rule 6: (13/1, lift 1.6)
##  trtbps <= 108
##  ->  class Alto riesgo  [0.867]
##
## Rule 7: (47/7, lift 1.5)
```

```
##  cp in {Angina atípica, Dolor no anginal}
##  caa <= 0
##  ->  class Alto riesgo  [0.837]
##
## Rule 8: (111/25, lift 1.4)
##  thall = Normal
##  ->  class Alto riesgo  [0.770]
##
## Default class: Alto riesgo
##
##
## Evaluation on training data (200 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       8   22(11.0%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##      77    15    (a): class Bajo riesgo
##       7   101    (b): class Alto riesgo
##
##
##  Attribute usage:
##
##   89.00% thall
##   65.00% caa
##   61.50% cp
##   44.50% trtbps
##   38.00% oldpeak
##   30.00% chol
##
##
## Time: 0.0 secs
```

Obtenemos la siguiente información del árbol de decisión:

El algoritmo ha obtenido 8 reglas de clasificación con un error del 11% y un acierto del 89%.

Analizaremos los mejores resultados:

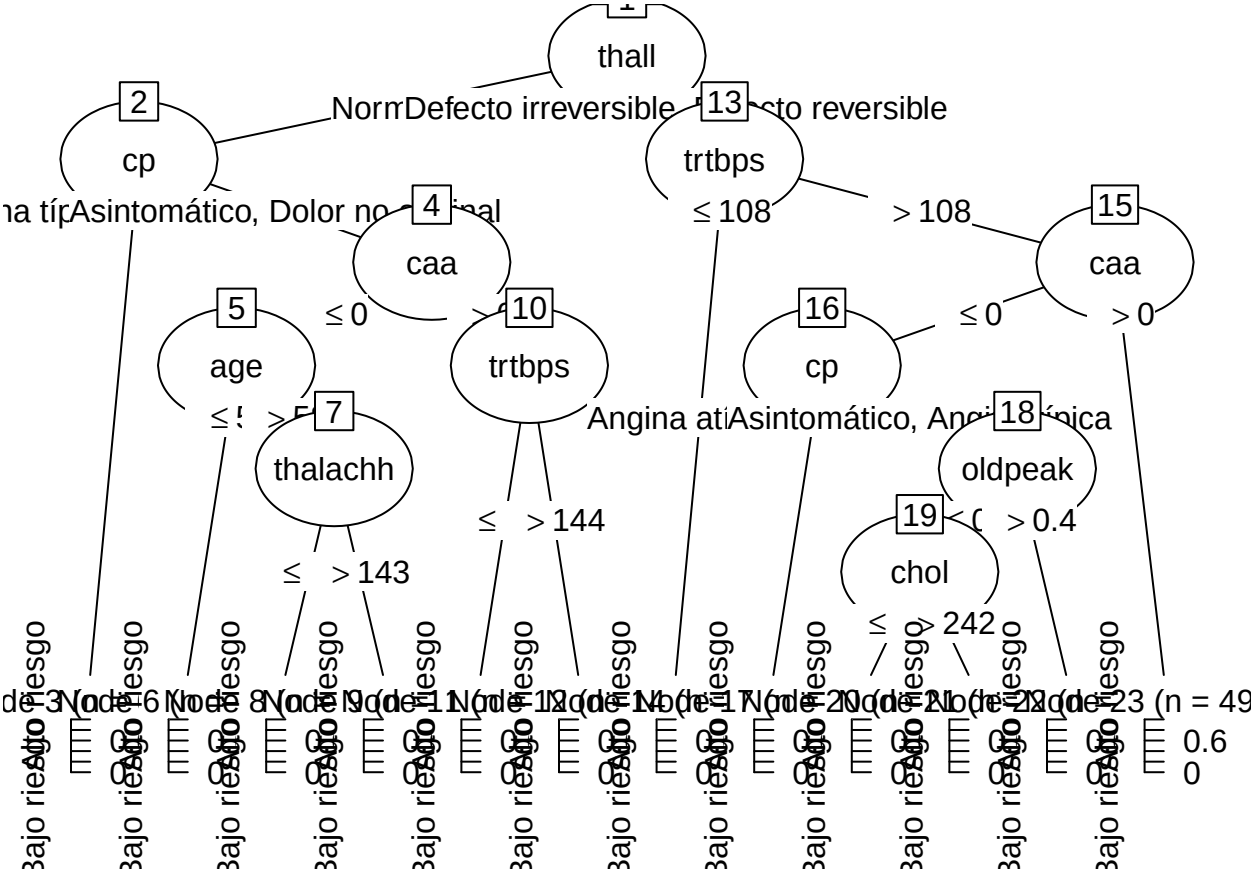
class Bajo riesgo 0.972: Persona con tipo de dolor en el pecho Asintomático o Angina típica, con la presión arterial en resposo > 108, con colesterol en sangre > 242 y con el resultado de la prueba de esfuerzo con Defecto irreversible o Defecto reversible tiene una probabilidad de no padecer una enfermedad cardíaca del 97%

class Alto riesgo 0.929: Persona con colesterol <= 242, con depresión en el segmento ST al hacer ejercicio en relación con el reposo <= 0.4 y con el número de vasos principales del corazón <= 0 tiene una probabilidad de padecer una enfermedad cardiaca del 93%.

Las variables más influyentes en la clasificación son: 89.00% thall, 65.00% caa, 61.50% cp, 44.50% trtbps, 38.00% oldpeak y 30.00% chol

Mostramos el árbol de manera gráfica.

```
mo_tree <- C50::C5.0(trainX, trainy)
plot(mo_tree)
```



Vamos a obtener la precisión del modelo a través de la matriz de confusión.

```
predicted_mo <- predict( mo, testX, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_mo == testy) / length(predicted_mo)
```

```
## [1] "La precisión del árbol es: 80.1980 %"
```

```
matriz_conf<-table(testy,Predicted=predicted_mo)
matriz_conf
```

```
##              Predicted
## testy         Bajo riesgo Alto riesgo
##   Bajo riesgo          32          13
##   Alto riesgo           7          49
```

El estudio de los datos nos muestra que $32 + 49 = 81$ has sido clasificados correctamente y $7 + 13 = 20$ no. Tenemos un 81% de aciertos, un porcentaje de acierto ligeramente superior al conjunto de entrenamiento.

```
porcentaje_correct<-100 * sum(diag(matriz_conf)) / sum(matriz_conf)
print(sprintf("El %% de registros correctamente clasificados es: %.4f %%",
              porcentaje_correct))
```

```
## [1] "El % de registros correctamente clasificados es: 80.1980 %"
```

El porcentaje de la matriz de confusión es igual al que hemos obtenido de las reglas de decisión.

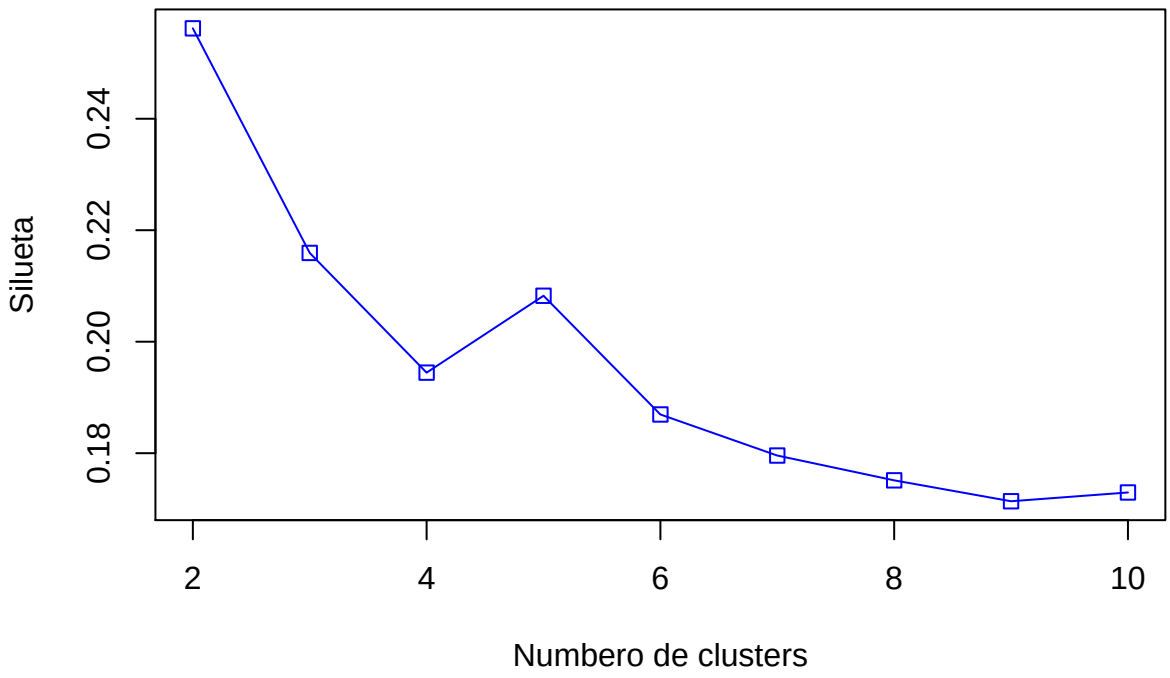
2.4.2.4 K-means K-means es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano.

K-means no funciona con datos categóricos así que tendremos que hacer un subconjunto con los datos numéricos del dataset, aplicaremos algoritmos de normalizar ya que este algoritmo es muy sensible a los outliers.

```
library(dplyr)
# Creamos el dataset con las variables numéricas
heart_km <- select_if(heartdata, is.numeric)

# Aplicamos algoritmo de normalizar para minimizar el efecto de los outliers.
normalizar <- function(x){
 return ((x-min(x))/(max(x)-min(x)))
}

heart_km$age <- normalizar(heart_km$age)
heart_km$trtbps   <- normalizar(heart_km$trtbps)
heart_km$thalachh <- normalizar(heart_km$thalachh)
heart_km$chol <- normalizar(heart_km$chol)
heart_km$oldpeak <- normalizar(heart_km$oldpeak)

# Elimimas caa ya que es categórica y no numérica.
heart_km$caa <- NULL
```

Al no conocer el número de clústers optimo, probaremos con diferentes valores y métodos.

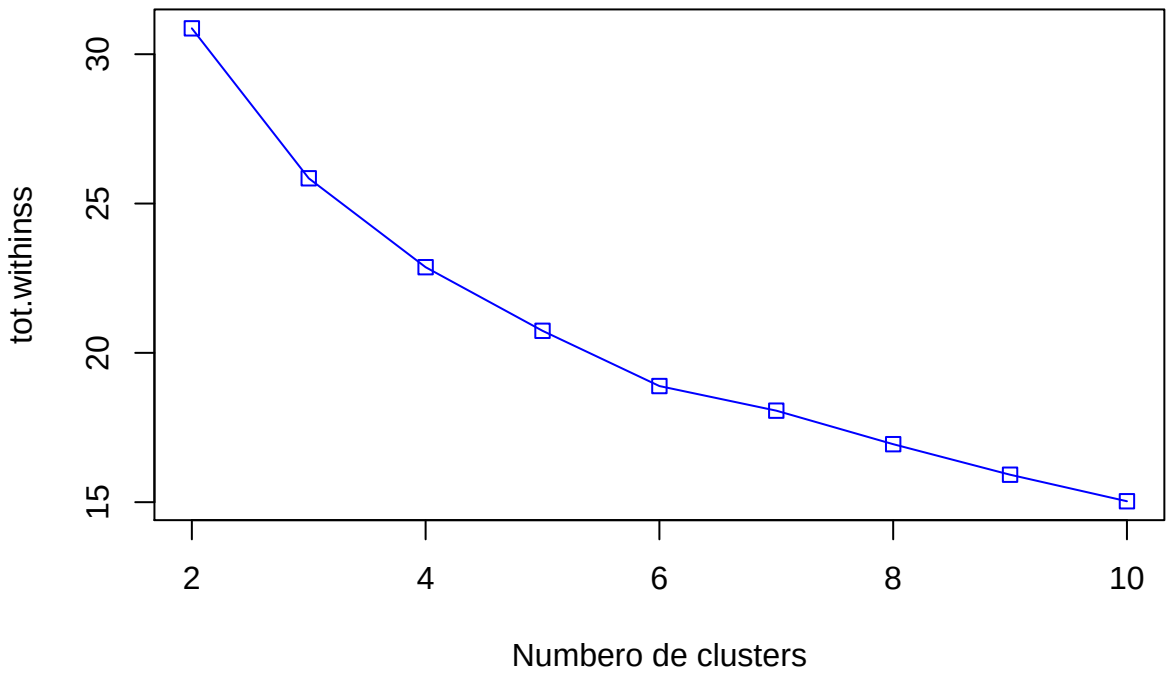
```
library(cluster)
d <- daisy(heart_km)
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(heart_km, i)
  y_cluster     <- fit$cluster
  sk            <- silhouette(y_cluster, d)
  resultados[i] <- mean(sk[,3])
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Numbero de clusters",
     ylab="Silueta")
```



La gráfica nos muestra que el número de cluster optimo es 2, vamos a seguir probando otros algoritmos para comprobar si realmente k=2.

Vamos aplicar el método elbow(codo). Este método utiliza los valores de la inercia obtenidos tras aplicar el K-means a diferente número de Clusters (desde 1 a N Clusters), siendo la inercia la suma de las distancias al cuadrado de cada objeto del Cluster a su centroide.

```
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(heart_km, i)
  resultados[i] <- fit$tot.withinss
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Numbero de clusters",
     ylab="tot.withinss")
```



La gráfica nos muestra una estabilización de la curva en k= 4.

Aplicamos el algoritmo de silueta media y Calinski-Harabasz.

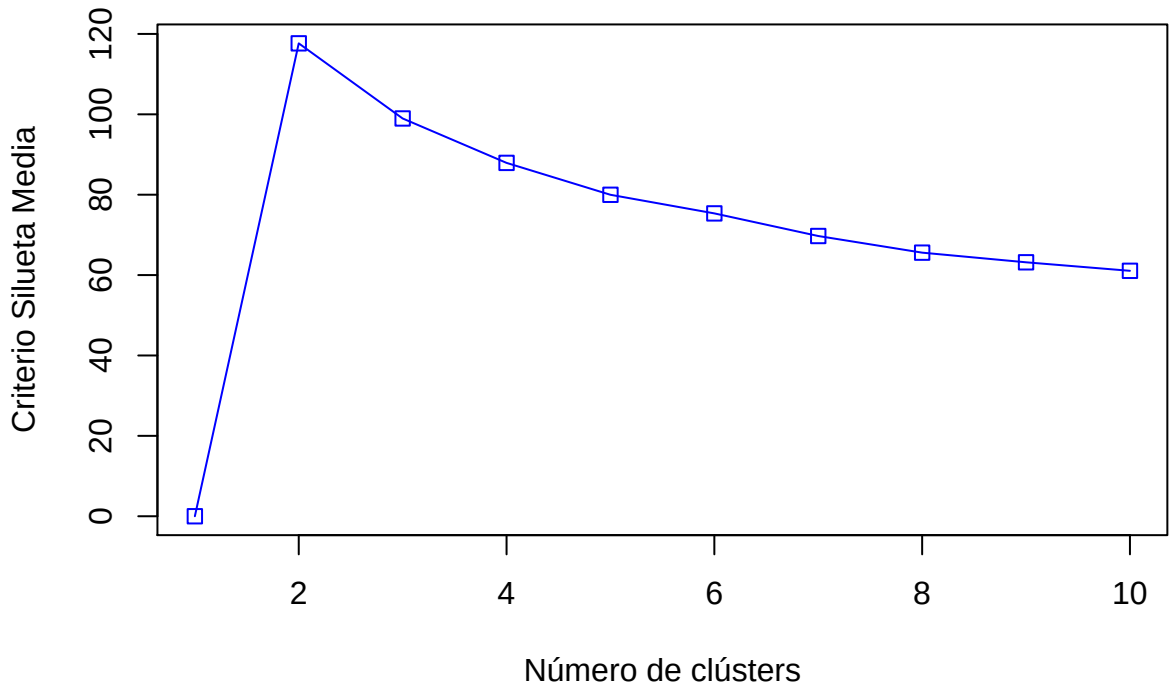
```
library(fpc)

fit_ch <- kmeansruns(heart_km, krange = 1:10, criterion = "ch")
fit_asw <- kmeansruns(heart_km, krange = 1:10, criterion = "asw")
fit_ch$bestk
```

```
## [1] 2
```

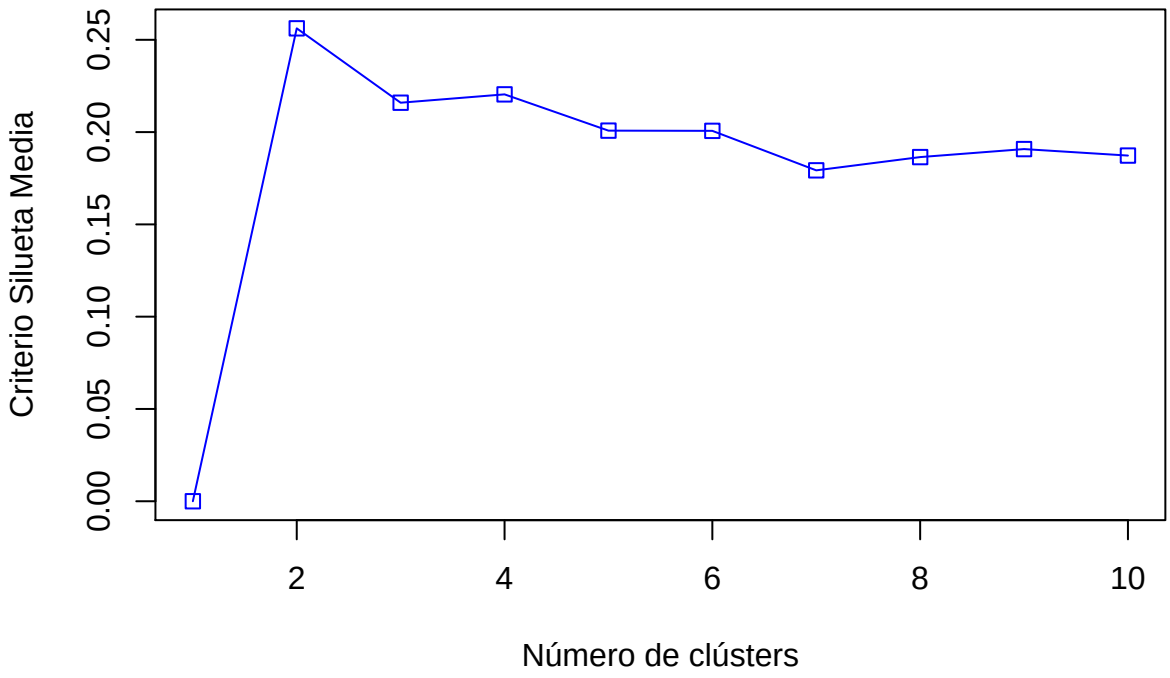
```
fit_asw$bestk
```

```
## [1] 2
```

```
plot(1:10,fit_ch$crit, type = "o", col="blue", pch=0, xlab="Número de clústers",
     ylab="Criterio Silueta Media")
```

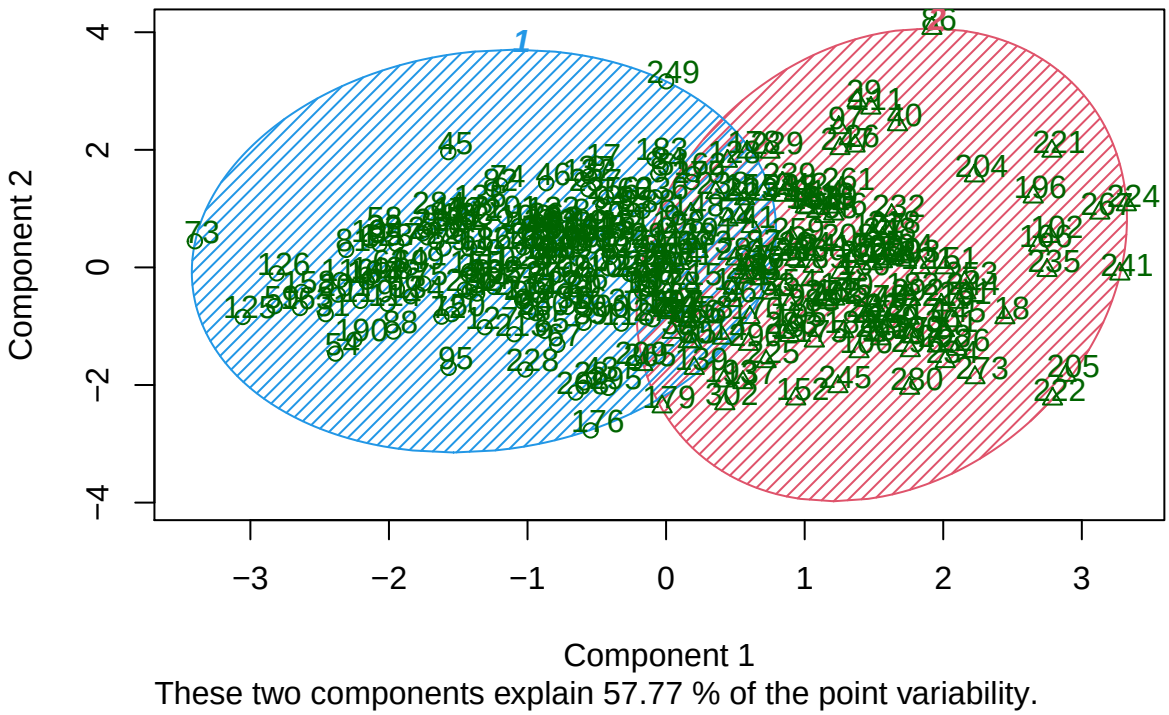



```
plot(1:10,fit_asw$crit, type = "o", col="blue", pch=0, xlab="Número de clústers"
     , ylab="Criterio Silueta Media")
```



Los dos métodos nos han dado el mismo resultado k=2.

```
# visualización de 2 clusters
km.res2 <- kmeans(heart_km, 2, nstart = 25)
clusplot(heart_km, km.res2$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



El algoritmo silhouette es 2

El algoritmo elbow(codo) es 4

El algoritmo silueta media es 2

El algoritmo de Calinski-Harabasz es 2

El resultado optimo es k=2, lo que concuerda con los datos que tenemos en el dataset, ya que puedes o no padecer enfermedad cardíaca. Esto nos confirma que la categoría de la variable target (Bajo riesgo, Alto riesgo) es correcto.

Vamos a estudiar la precisión del modelo para k =2

```
table(km.res2$cluster,heartdata$output)
```

```
##
##     Bajo riesgo Alto riesgo
##   1          38         119
##   2          99          45
```

```
100*((99+119)/(218+38+45))
```

```
## [1] 72.42525
```

Tenemos un total de instancias correctamente clasificadas de 218, el total de casos con error de clasificación son 85, por lo que podemos calcular: exactitud = 218/(218+83) = 0,7242, con lo que el modelo para k=2 tienen una exactitud del 72%.

2.5 Conclusión

Inicialmente hemos realizado un proceso de limpieza y preprocesado de los datos del dataset que estamos estudiando, el cual contiene datos reales sobre pruebas médicas para el diagnóstico de enfermedades cardíacas. En este proceso hemos cambiado los nombres a las variables y hemos estudiado los valores extremos(Outliers).

Seguidamente hicimos un análisis exploratorio del conjunto de variables que componen el dataset para tener una primera aproximación de su comportamiento, y así poder hacernos una idea de como se comportarán al aplicar algoritmos estadísticos.

Aplicamos un algoritmo de regresión lineal que es una buena manera de conocer la relación que existe entre las variables del modelo, seguidamente aplicamos el algoritmo supervisado de arbole de decisión que nos sirve para sabes los diferentes casos (reglas de comportamiento) de la variable objetivo (Output)

Para finalizar, aplicamos un algoritmo no supervisado de clasificación (K-means) el cual nos dió como resultado 2 clusters tal y como se agrupa la variable objetivo(output) (Bajo riesgo, Alto riesgo).

Cada uno de los pasos realizados en este estudios nos han ayudado a tener una profunda compresión de los datos que conforman el dataset. Por lo tanto, podemos afirmar, que hemos podido responder el problema de inicio, el cual consistía en poder responder mediante el análisis de datos si se podía diagnosticar una enfermedad cardíaca.

2.6 Recursos

Dr.Luis Azcona - El electrocardiograma. https://www.fbbva.es/microsites/salud_cardio/mult/fbbva_libroCorazon_cap4.pdf

Healthline.com - Thallium Stress Test. https://www.healthline.com/health/thallium-stress-test

---

2.7 Contribuciones/Firma integrantes

```
integrantes <- data.frame(Contribuciones=
   c("Investigación previa","Redacción de las respuestas","Desarrollo código"),
   Firma=c("Jose Antonio Jara Pérez / Óscar López Montero",
           "Jose Antonio Jara Pérez / Óscar López Montero",
           "Jose Antonio Jara Pérez / Óscar López Montero"))

integrantes
```

```
##               Contribuciones                                          Firma
## 1        Investigación previa Jose Antonio Jara Pérez / Óscar López Montero
## 2 Redacción de las respuestas Jose Antonio Jara Pérez / Óscar López Montero
## 3           Desarrollo código Jose Antonio Jara Pérez / Óscar López Montero
```