Práctica 2 - Limpieza y análisis de datos

Eric Serrulla

12/21/2019

Contents

1. Descripción del dataset

El dataset escogido **Heart Desease UCI** se ha descargado de Kaggle y corresponde a los resultados de pruebas sobre pacientes que presentan o no enfermedades cardiovasculares. Es el resultado de 4 bases de datos provinientes de Cleveland, Hungría, Suiza y VA Long Beach.
Este dataset es interesate porque permite conocer relaciones entre distintas variables que condicionan las enfermedades cardiovasculares.
Puede usarse para realizar estudios medicinales, para predecir potenciales enfermedades cardiovasculares o encontrar patrones en los síntomas, por ejemplo. Con este dataset se pretende responder a preguntas como *¿Se incrementa el riesgo de padecer enferemdades cardiovasculares según el género de la persona? ¿Cómo varían los síntomas según la edad y otras medidas relativas al sistema cardiovascular?.*

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

El fichero de datos contiene un total de 303 registros para 14 variables. Estas variables son:

- **age**. Edad de la persona en años
- **sex**. Género: 1 = masculino; 0 = femenino
- **cp**. Tipo de dolor en el pecho (Del 1 al 4).
    1. Angina típica
    2. Angina atípica
    3. Dolor no anginal
    4. Asintomático
- **trestbps**. Presión arterial en reposo, en mm Hg, al ingresar en el hospital.
- **chol**. Colesterol sérico en mg/dl.
- **fbs**. Si el azúcar en sangre en ayunas es mayor a 120 mg/dl. 1 = verdadero; 0 = falso.
- **restecg**. Resultados electrocardiográficos en reposo (Del 0 al 2).
    0. Normal
    1. Anormalidad de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST > 0.05 mV)
    2. Muestra probable o definitiva hipertrofia ventricular izquierda según el *Estes' criteria.*

- **thalach**. Máxima frecuencia cardíaca alcanzada.
- **exang**. Angina inducida por ejercicio. 1 = verdadero; 0 = falso.
- **oldpeak**. Depresión del ST inducida por el ejercicio relativo al descanso.
- **slope**. Pendiente de ST en el pico de ejercicio (Del 1 al 3).
    1. Ascendiente
    2. Plana
    3. Descendente
- **ca**. Número de vasos principales (0-3) coloreados por fluorospía.
- **thal**. THAL. (Del 1 al 3).
    1. Defecto fijo
    2. Normal
    3. Defecto reversible
- **target**. 0 = No tiene enfermedad cardiovascular. 1 = Tiene enfermedad cardiovascular.

3. Limpieza de los datos

3.1. Elementos vacíos

```
heartData <- read.csv('heart.csv')
head(heartData)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3  41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

```
summary(heartData)
```

```
##       age             sex               cp           trestbps
##  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          thalach
##  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
##      exang           oldpeak         slope             ca
##  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
##       thal           target
##  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
```

Encontramos que en el caso de la variable `thal`, hay 2 registros con valor fuera de rango, es decir, con valor 0. Gracias a la documentación del repositorio sabemos que estos valores corresponden a datos desconocidos, que

en el dataset original habían sido marcados con un "?". Lo mismo pasa con las filas de `ca` que contienen valor 4, fuera del rango 0-3. Para tratarlos como elementos vacíos los pasamos a tipo NA.

```
emptyCa <- which(heartData$ca > 3)
heartData[emptyCa,]$ca <- NA
emptyThal <- which(heartData$thal == 0)
heartData[emptyThal,]$thal <- NA
```

Solucionados estos valores, corregimos las variables a numéricas y factor, para así poder trabajar con ellas en su formato correcto.

```
heartData$age <- as.numeric(heartData$age)
heartData$trestbps <- as.numeric(heartData$trestbps)
heartData$chol <- as.numeric(heartData$chol)
heartData$thalach <- as.numeric(heartData$thalach)
heartData$sex <- factor(heartData$sex)
levels(heartData$sex) <- c("female", "male")
heartData$cp <- factor(heartData$cp)
levels(heartData$cp) <- c("typical","atypical","non-anginal","asymptomatic")
heartData$fbs <- factor(heartData$fbs)
levels(heartData$fbs) <- c("false", "true")
heartData$restecg <- factor(heartData$restecg)
levels(heartData$restecg) <- c("normal","stt","hypertrophy")
heartData$exang <- factor(heartData$exang)
levels(heartData$exang) <- c("no","yes")
heartData$slope <- factor(heartData$slope)
levels(heartData$slope) <- c("upsloping","flat","downsloping")
heartData$ca <- factor(heartData$ca)
heartData$thal <- factor(heartData$thal)
levels(heartData$thal) <- c("normal","fixed","reversable")
heartData$target <- factor(heartData$target)
levels(heartData$target) <- c("no","yes")

summary(heartData)
```

```
##       age            sex                 cp         trestbps
##  Min.   :29.00   female: 96   typical     :143   Min.   : 94.0
##  1st Qu.:47.50   male  :207   atypical    : 50   1st Qu.:120.0
##  Median :55.00                non-anginal : 87   Median :130.0
##  Mean   :54.37                asymptomatic: 23   Mean   :131.6
##  3rd Qu.:61.00                                   3rd Qu.:140.0
##  Max.   :77.00                                   Max.   :200.0
##       chol          fbs             restecg       thalach      exang
##  Min.   :126.0   false:258   normal     :147   Min.   : 71.0   no :204
##  1st Qu.:211.0   true : 45   stt        :152   1st Qu.:133.5   yes: 99
##  Median :240.0               hypertrophy:  4   Median :153.0
##  Mean   :246.3                                 Mean   :149.6
##  3rd Qu.:274.5                                 3rd Qu.:166.0
##  Max.   :564.0                                 Max.   :202.0
##     oldpeak             slope        ca              thal     target
##  Min.   :0.00   upsloping  : 21   0   :175   normal    : 18   no :138
##  1st Qu.:0.00   flat       :140   1   : 65   fixed     :166   yes:165
##  Median :0.80   downsloping:142   2   : 38   reversable:117
##  Mean   :1.04                     3   : 20   NA's      :  2
##  3rd Qu.:1.60                     NA's:  5
```

```
##  Max.   :6.20
```

Para imputar el valor de los `ca` y `thal` perdidos lo haremos mediante el método de k-vecinos más próximos, (**kNN-imputation**). Para ello utilizaremos la librería VIM.

```
table(heartData$ca) #Previsualización de los valores de ca
```

```
##
##   0   1   2   3
## 175  65  38  20
```

```
table(heartData$thal) #Previsualización de los valores de thal
```

```
##
##     normal      fixed reversable
##         18        166        117
```

```
suppressWarnings(suppressMessages(library(VIM)))
heartData$ca <- kNN(heartData)$ca
heartData$thal <- kNN(heartData)$thal

table(heartData$ca)  #Previsualización de los valores de ca tras imputación
```

```
##
##   0   1   2   3
## 180  65  38  20
```

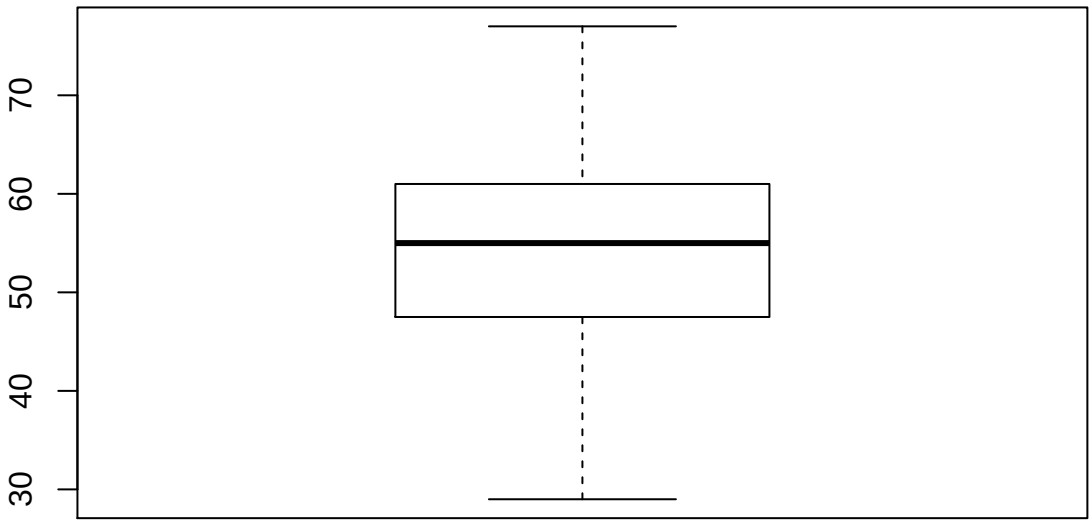
```
table(heartData$thal) #Previsualización de los valores de thal tras imputación
```

```
##
##     normal      fixed reversable
##         18        167        118
```

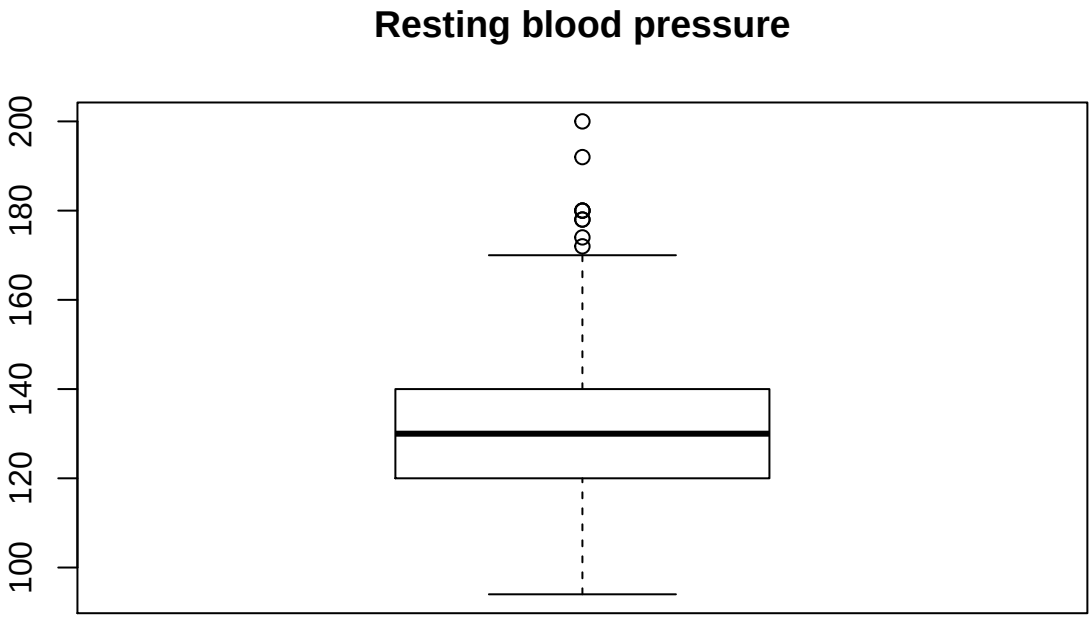
3.2. Identificación y tratamiento de valores extremos

Comprobamos mediante el diagrama de cajas la distribución de las variables para encontrar posibles valores extremos.

```
boxplot(heartData$age)
title("Age")
```

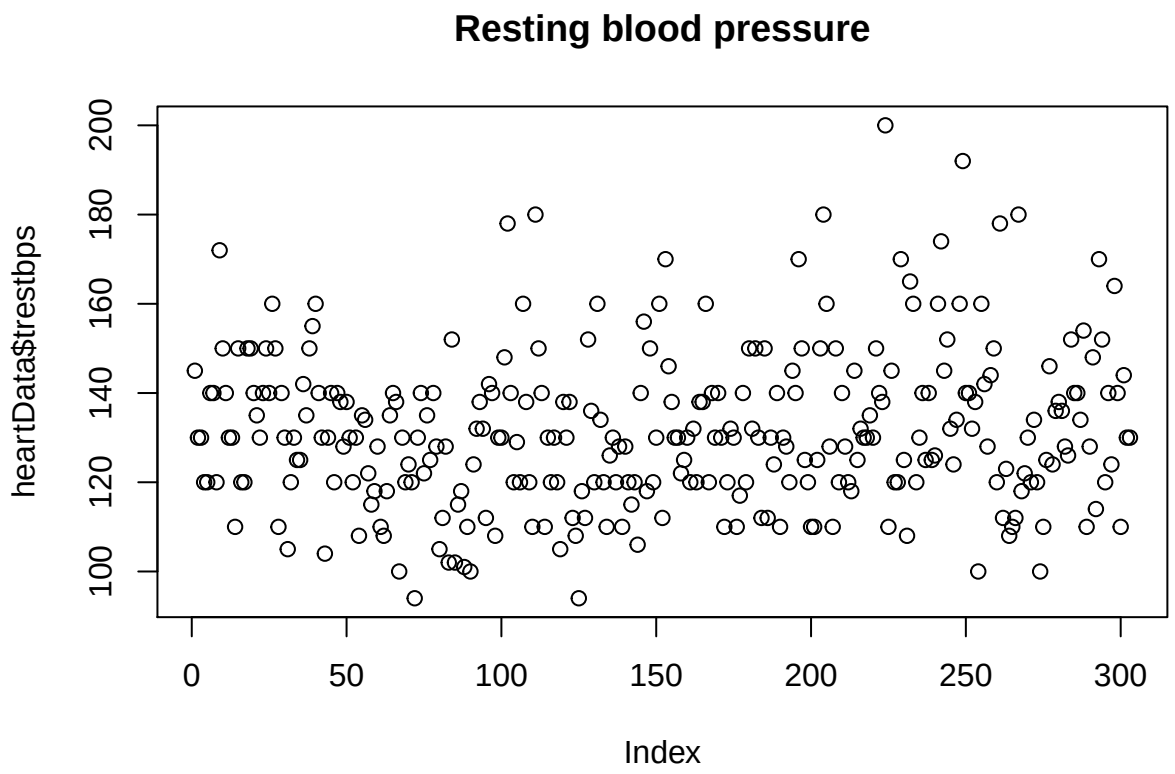



```
boxplot(heartData$trestbps)
title("Resting blood pressure")
```



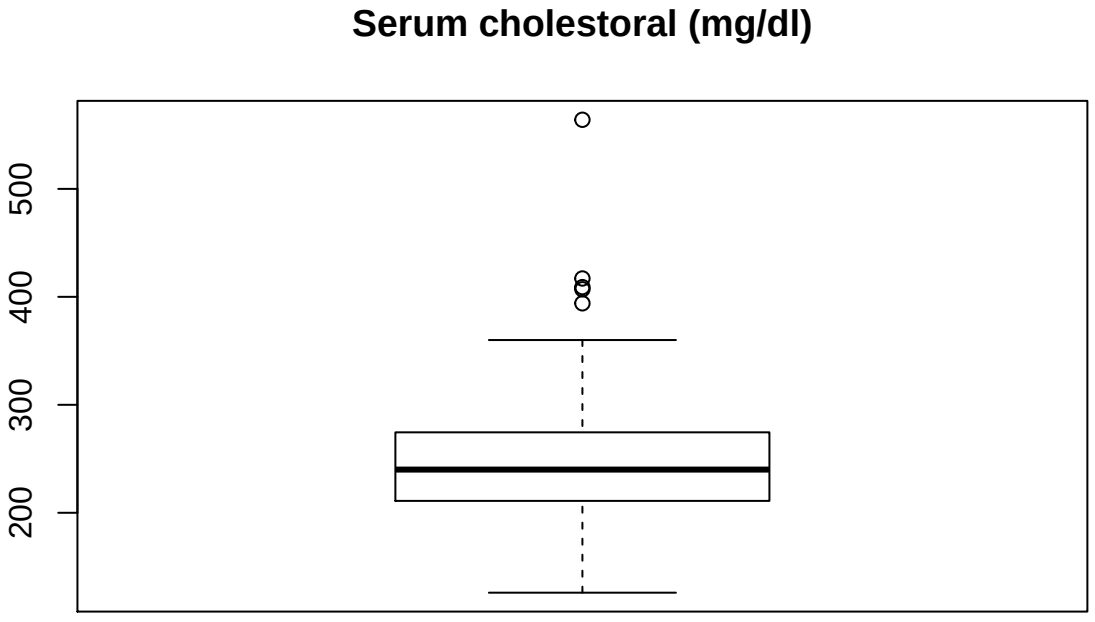
En la presión arterial en reposo, los valores fuera y alejados de la caja no se tratan de valores extremos que se traduzcan en error de inserción. Si lo vemos en un gráfico plot, se entiende mejor.

```
plot(heartData$trestbps)
title("Resting blood pressure")
```

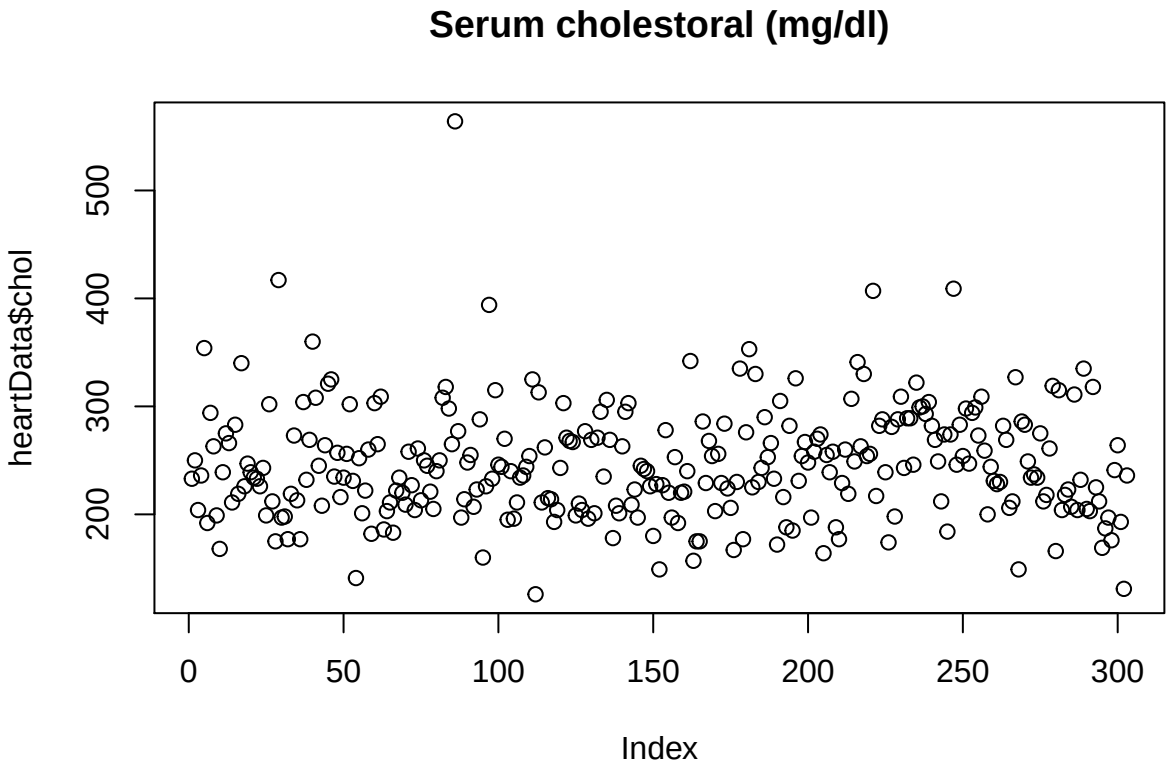


En este estudio nos interesa encontrar esos picos en los niveles de presión arterial, ya que se salen del rango recomendable, y lo mismo si fueran demasiado bajos.

```
boxplot(heartData$chol)
title("Serum cholestoral (mg/dl)")
```




```
plot(heartData$chol)
title("Serum cholestoral (mg/dl)")
```

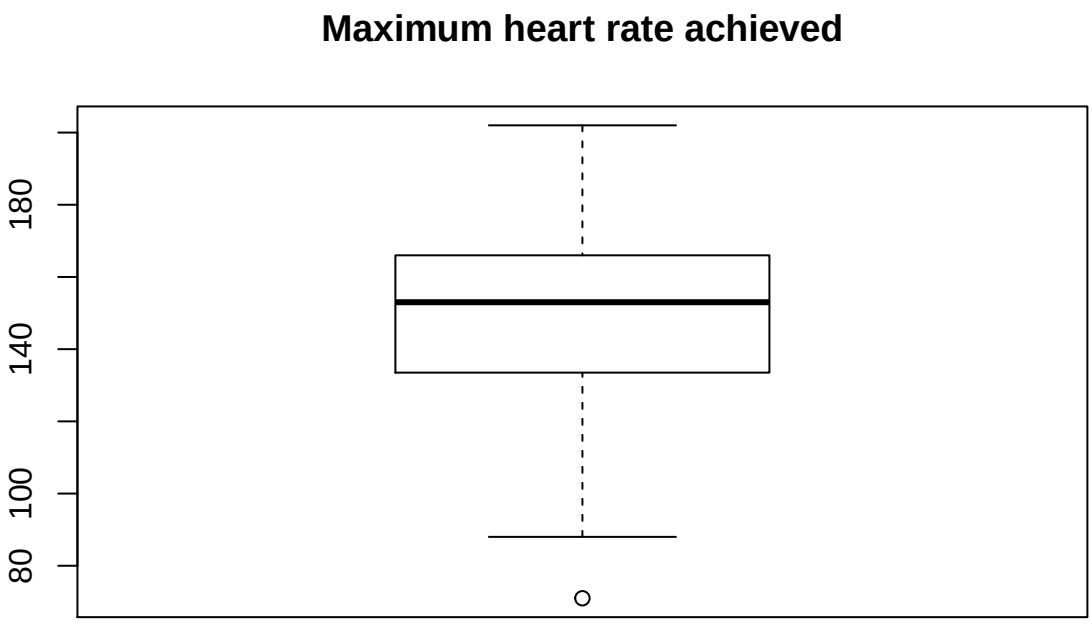


En los niveles de colesterol encontramos un caso donde el valor de estos son 564. Aunque es un valor muy alto, tiene sentido ya que esta persona indica enfermedad (valor target = yes).

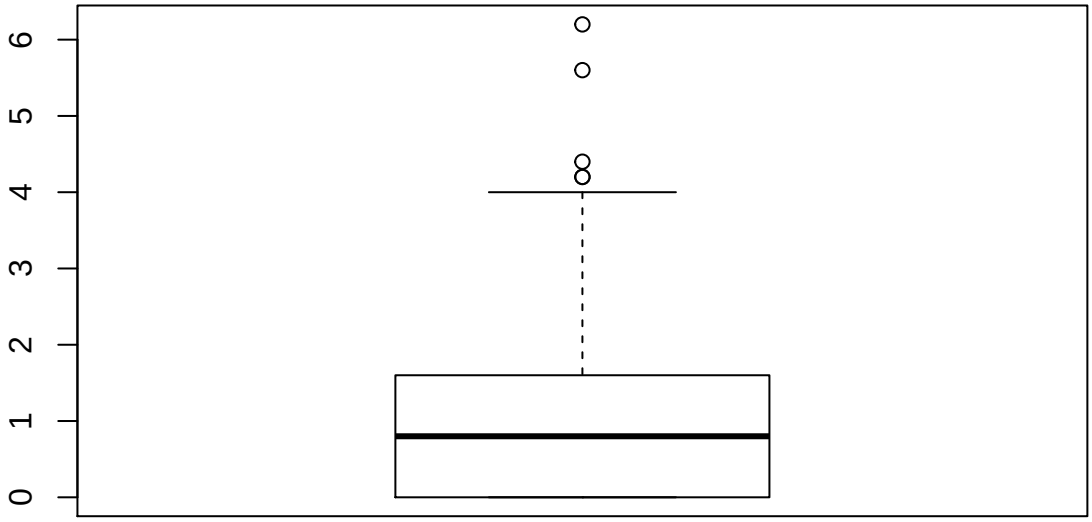
```
heartData[which(heartData$chol>500),]
```

```
##    age    sex          cp trestbps chol   fbs restecg thalach exang
## 86  67 female non-anginal      115  564 false  normal     160    no
##    oldpeak slope ca       thal target
## 86     1.6  flat  0 reversable    yes
```

```
boxplot(heartData$thalach)
title("Maximum heart rate achieved")
```




```
boxplot(heartData$oldpeak)
title("ST depression")
```

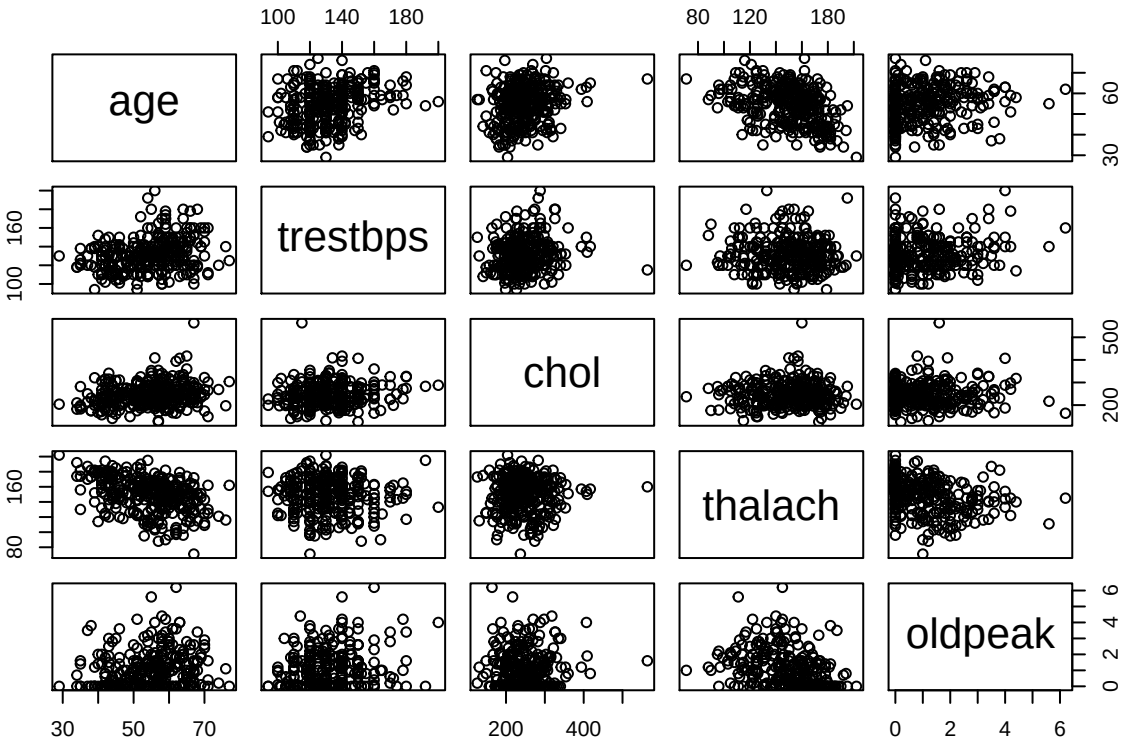


4. Análisis de los datos

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

Antes de comenzar seleccionaremos las variables numéricas y visualizamos si hay relación entre ellas.

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



Aparentemente no podemos afirmar que hay relación directa entre dos variables, ya que los gráficos no lo indican con suficiente definición. Por ello, pasamos a observar los resultados del **coeficiente de correlación** mediante la **prueba de Spearman**.

```
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 "age"
for (i in 2:(ncol(heartData.numeric))) {
  spearman_test = cor.test(heartData.numeric[,i],
                           heartData.numeric[,1],
                           method = "spearman",
                           exact = FALSE)
  corr_coef = spearman_test$estimate
  p_val = spearman_test$p.value

  # Se añade una fila nueva a la matriz
  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(heartData.numeric)[i]
}

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

```
##             estimate      p-value
## trestbps  0.2856168 4.261709e-07
## chol      0.1957860 6.099143e-04
## thalach  -0.3980524 6.024321e-13
## oldpeak   0.2682912 2.158797e-06
```

El valor más alto del coeficiente de correlación (`0.398`) que encontramos respecto a la edad es con la variable de la **máxima frecuencia cardíaca alcanzada**, y es de tipo inversamente proporcional. Si observamos el gráfico anterior, a pesar de la amplia dispersión, se aprecia una ligera pendiente negativa.

Con esta información, formulamos las pruebas que queremos llevar a cabo:

- Sexo y resultado en la prueba. ¿Son los hombres són más propensos a sufrir enfermedades cardiovasculares respecto a las mujeres?
- Influencia de la edad y resultado en la prueba. Predecir la frecuencia cardíaca máxima según la edad del paciente.
- Predicción del diagnóstico de enfermedad teniendo en cuenta los resultados en las pruebas y el perfil del paciente.

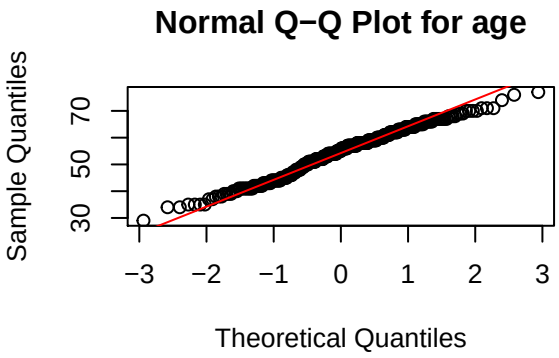
Para los análisis que queremos realizar es necesario disponer de todas las columnas de datos.

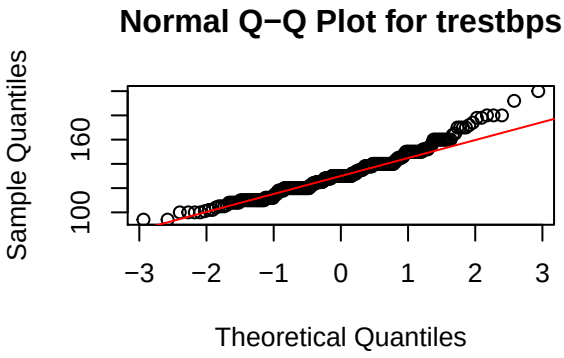
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

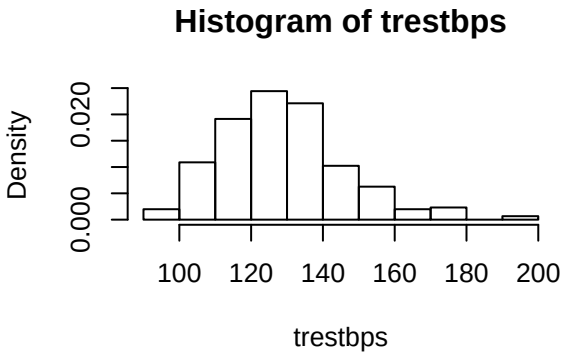
Para comprobar si podemos asumir normalidad en la distribución de la muestra, observamos el **gráfico de cuantiles** y el **histograma** de cada una de las variables numéricas, además de aplicar el **test de Shapiro-Wilk**.

```
par(mfrow=c(2,2))
for(i in 1:ncol(heartData.numeric)) {
  title <- colnames(heartData.numeric[i])
  qqnorm(heartData.numeric[,i],main = paste("Normal Q-Q Plot for",title))
  qqline(heartData.numeric[,i],col="red")
  hist(heartData.numeric[,i],
        xlab= title,
        freq = FALSE,
        main = paste("Histogram of", title))
  shapiro.test(heartData.numeric[,i])
  print(paste("Test for", title))
  print(shapiro.test(heartData.numeric[,i]))
}
```

```
## [1] "Test for age"
##
##  Shapiro-Wilk normality test
##
## data:  heartData.numeric[, i]
## W = 0.98637, p-value = 0.005798
```

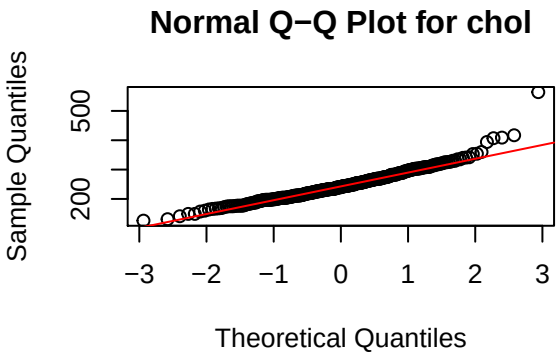


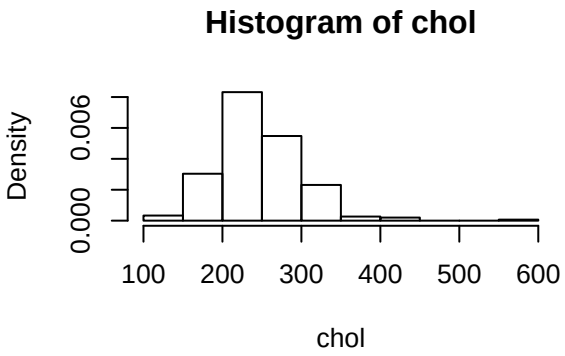


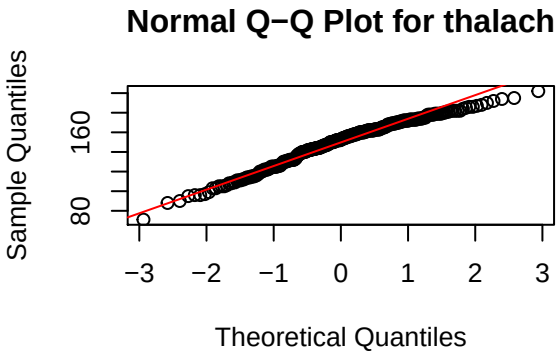


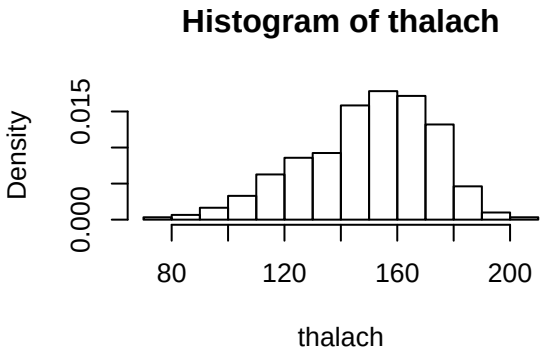

```
## [1] "Test for trestbps"
##
##  Shapiro-Wilk normality test
##
## data:  heartData.numeric[, i]
## W = 0.96592, p-value = 1.458e-06
```

```
## [1] "Test for chol"
##
##  Shapiro-Wilk normality test
##
## data:  heartData.numeric[, i]
## W = 0.94688, p-value = 5.365e-09
```



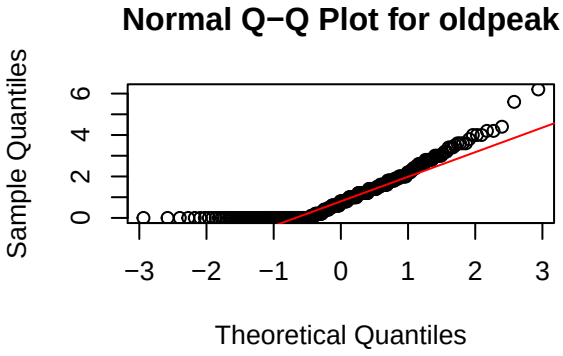


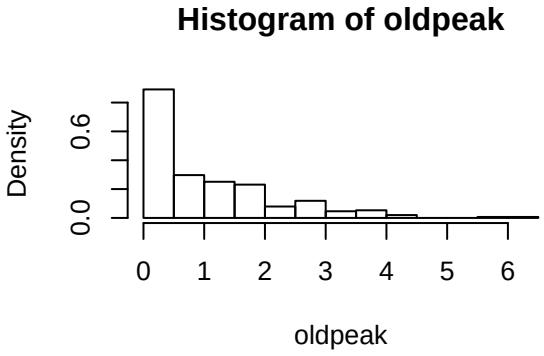





```
## [1] "Test for thalach"
##
##  Shapiro-Wilk normality test
##
## data:  heartData.numeric[, i]
## W = 0.97632, p-value = 6.621e-05
```

```
## [1] "Test for oldpeak"
##
##  Shapiro-Wilk normality test
##
## data:  heartData.numeric[, i]
## W = 0.84418, p-value < 2.2e-16
```





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 esta homogeneidad en cuanto al colesterol los grupos conformados por mujeres frente a los hombres.

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

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

Para un nivel de confianza del 95% 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.

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

4.3.1. Contraste de hipótesis sobre el colesterol en hombres y mujeres

Lo primero será hacer un contraste de hipótesis sobre dos muestras para determinar si los niveles de colesterol en mujeres son los mismos que en hombres o son mayores. Para ello formulamos bajo un nivel de confianza del 95%, la **hipótesis nula** H_0 y la **alternativa** H_1, que comparan la diferencia entre las dos medias μ_1, μ_2 correspondientes a las muestras de cada sexo:

$H_0 : \mu_1 - \mu_2 = 0$
$H_1 : \mu_1 - \mu_2 > 0$

```
t.test(heartData[heartData$sex == "female",]$chol, heartData[heartData$sex == "male",]$chol)
```

```
##
##  Welch Two Sample t-test
##
## data:  heartData[heartData$sex == "female", ]$chol and heartData[heartData$sex == "male", ]$chol
## t = 3.0244, df = 134.39, p-value = 0.002985
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   7.617474 36.406982
## sample estimates:
## mean of x mean of y
##  261.3021  239.2899
```

Para un nivel de confianza del 95% obtenemos un p-valor de 0.003, inferior al nivel de confianza 0.05, por lo tanto rechazamos la hipótesis de que la media de niveles de colesterol entre hombres y mujeres es igual. En su lugar aceptamos la hipótesis alternativa de que la media de niveles de colesterol en mujeres es **superior** a la de los de los hombres.

4.3.2. Modelo de regresión lineal

Tras la visualización de correlaciones en el apartado 4.1, procedemos a crear un modelo de regresión lineal simple con la **frecuencia cardíaca máxima alcanzada** como variable dependiente y la **edad** como independiente.

```
model <- lm(thalach ~ age, data = heartData.numeric)
summary(model)
```

```
##
## Call:
## lm(formula = thalach ~ age, data = heartData.numeric)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -65.949 -11.954   3.975  15.921  44.985
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 204.2892     7.3485  27.800  < 2e-16 ***
## age          -1.0051     0.1333  -7.539 5.63e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.04 on 301 degrees of freedom
## Multiple R-squared:  0.1588, Adjusted R-squared:  0.156
## F-statistic: 56.83 on 1 and 301 DF,  p-value: 5.628e-13
```

El valor de ***R-squared*** es muy bajo, por lo que no es un modelo óptimo. Probamos a añadir más variables independientes para ver cómo cambia este valor.

```
model <- lm(thalach ~ ., data = heartData.numeric)
summary(model)
```

```
##
## Call:
## lm(formula = thalach ~ ., data = heartData.numeric)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.714 -11.864   3.267  13.882  39.837
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 181.67887   10.63340  17.086  < 2e-16 ***
## age          -0.96792    0.13630  -7.101 9.11e-12 ***
## trestbps      0.13980    0.06927   2.018   0.0445 *
## chol          0.03289    0.02280   1.443   0.1502
## oldpeak      -5.68717    1.02525  -5.547 6.41e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.02 on 298 degrees of freedom
## Multiple R-squared:  0.2464, Adjusted R-squared:  0.2363
## F-statistic: 24.36 on 4 and 298 DF,  p-value: < 2.2e-16
```

Con todas las demás variables numéricas como independientes, el modelo mejora hasta un `0.246`, aunque aún está lejos del que consideraríamos un modelo ideal.
Utilizando este modelo, pasamos a predecir los valores de `thalach` y comparar los resultados con los reales. Para ello, dedicaremos un 80% del dataset para entrenar el modelo y un 20% para probarlo.

```
# Predicción de thalach para pacientes con enfermedad cardiovascular
heartData.glm<-heartData[which(heartData$target=="yes"),]
ntrain <- nrow(heartData.glm)*0.8
ntest <- nrow(heartData.glm)*0.2
index_train<-sample(1:nrow(heartData.glm),size = ntrain)
train<-heartData.glm[index_train,]
test<-heartData.glm[-index_train,]
```

```r
model<-lm(thalach ~ age, data=train)
summary(model)
```

```
##
## Call:
## lm(formula = thalach ~ age, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.688  -7.241   1.818   9.490  30.176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 219.2077     7.3937   29.65  < 2e-16 ***
## age          -1.1420     0.1406   -8.12 3.12e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.21 on 130 degrees of freedom
## Multiple R-squared:  0.3365, Adjusted R-squared:  0.3314
## F-statistic: 65.93 on 1 and 130 DF,  p-value: 3.122e-13
```

```r
predicted.data<-predict(model, test, type="response")
mc_sl<-data.frame(
  real=test$thalach,
  predicted= predicted.data,
  dif=ifelse(test$thalach>predicted.data, -predicted.data*100/test$thalach,predicted.data*100/test$thala
)
colnames(mc_sl)<-c("Real","Predicho","Dif (%)")
kable(mc_sl)
```

	Real	Predicho	Dif (%)
1	150	147.2619	-98.17457
17	172	152.9718	-88.93712
24	137	149.5459	109.15756
28	123	160.9658	130.86653
35	125	160.9658	128.77267
40	151	144.9779	-96.01183
44	143	158.6818	110.96632
48	156	165.5338	106.11143
50	160	158.6818	-99.17615
57	186	164.3918	-88.38270
59	174	180.3798	103.66655
61	130	138.1259	106.25067
72	154	160.9658	104.52327
78	164	151.8298	-92.57918
85	122	171.2438	140.36378
86	160	142.6939	-89.18367
90	122	152.9718	125.38676
93	169	159.8238	-94.57032
95	138	167.8178	121.60712
97	157	148.4039	-94.52475
103	179	147.2619	-82.26919
118	162	155.2558	-95.83694

	Real	Predicho	Dif (%)
122	182	151.8298	-83.42299
124	167	157.5398	-94.33523
140	105	146.1199	139.16177
143	173	171.2438	-98.98486
144	142	142.6939	100.48864
149	169	168.9598	-99.97622
151	138	143.8359	104.22889
152	125	138.1259	110.50070
153	155	146.1199	-94.27088
155	152	174.6698	114.91435
162	166	156.3978	-94.21557

Como era de esperar, el modelo se aproxima bastante bien en algunos casos pero en general se desvía demasiado, por lo que no es fiable si lo que buscamos es obtener valores muy aproximados a la realidad.

4.3.3. Modelo de regresión logística

Para estimar si el paciente sufre enfermedad cardiovascular o no, creamos un modelo de regresión logística y observamos qué variables tienen más peso a la hora de estimar el diagnóstico.

```
glmodel <- glm(target ~ ., data = heartData, family = "binomial")
summary(glmodel)
```

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = heartData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9386  -0.2784   0.1120   0.4426   3.0966
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.792111   2.964847   0.942 0.346327
## age                  0.027111   0.025418   1.067 0.286146
## sexmale             -1.801163   0.568664  -3.167 0.001538 **
## cpatypical           0.925167   0.575096   1.609 0.107678
## cpnon-anginal        1.980816   0.527290   3.757 0.000172 ***
## cpasymptomatic       2.385090   0.715078   3.335 0.000852 ***
## trestbps            -0.025928   0.011892  -2.180 0.029234 *
## chol                -0.004219   0.004216  -1.001 0.316885
## fbstrue              0.452039   0.578899   0.781 0.434885
## restecgstt           0.468147   0.400489   1.169 0.242428
## restecghypertrophy -0.675007   2.760567  -0.245 0.806830
## thalach              0.019719   0.011835   1.666 0.095674 .
## exangyes            -0.778251   0.450879  -1.726 0.084333 .
## oldpeak             -0.410335   0.241835  -1.697 0.089742 .
## slopeflat           -0.750241   0.880449  -0.852 0.394152
## slopedownsloping     0.652269   0.949390   0.687 0.492058
## ca1                 -2.338144   0.524557  -4.457 8.30e-06 ***
## ca2                 -3.448734   0.807630  -4.270 1.95e-05 ***
## ca3                 -2.248655   0.930588  -2.416 0.015676 *
## thalfixed           -0.233847   0.818238  -0.286 0.775036
```

```
## thalreversable     -1.673754   0.798578  -2.096 0.036089 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 417.64  on 302  degrees of freedom
## Residual deviance: 180.28  on 282  degrees of freedom
## AIC: 222.28
##
## Number of Fisher Scoring iterations: 6
```

Observamos que solo algunas de las variables son significantes para el modelo. Creamos una tabla con los valores de `target` y la probabilidad calculada por el modelo y ordenamos las filas por el valor de probabilidad ascendente.

```
predicted.data<-data.frame(
  real=heartData$target,
  predicted.probability= glmodel$fitted.values
)


predicted.data <- predicted.data[order(predicted.data$predicted.probability, decreasing=FALSE),]
predicted.data$rank <- 1:nrow(predicted.data)

kable(head(predicted.data))
```

	real	predicted.probability	rank
194	no	0.0006541	1
286	no	0.0007032	2
257	no	0.0007713	3
263	no	0.0008400	4
251	no	0.0010970	5
175	no	0.0012859	6

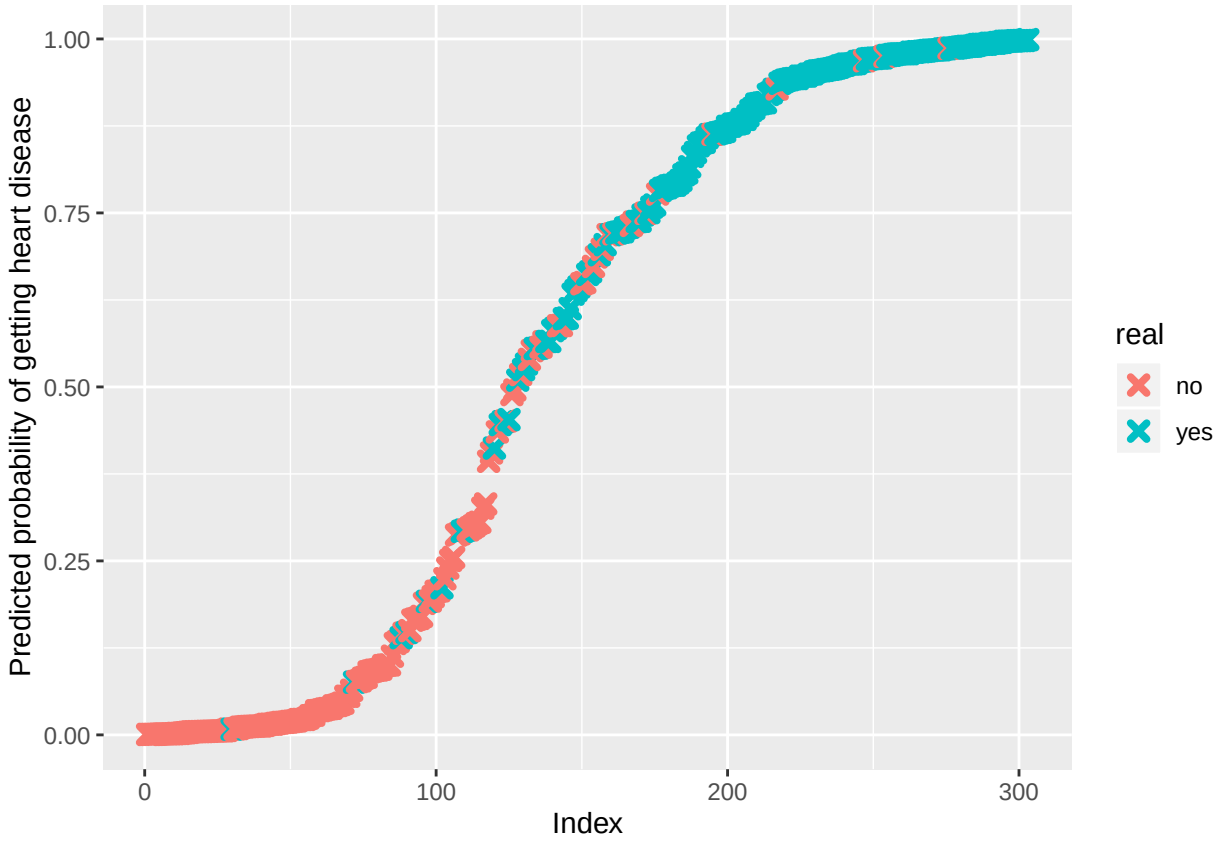
```
kable(tail(predicted.data))
```

	real	predicted.probability	rank
148	yes	0.9969553	298
116	yes	0.9975184	299
124	yes	0.9975803	300
17	yes	0.9977119	301
37	yes	0.9978601	302
125	yes	0.9990609	303

Podemos observar como los niveles bajos de probabilidad se corresponden con el valor `no` del atributo `target`, que describe si el paciente sufre o no enfermedad cardiovascular, mientras que los niveles altos indican el caso afirmativo. Por lo tanto, el modelo ha funcionado bastante bien. A continuación se muestra la misma información en formato gráfico:

```
library(ggplot2)
suppressWarnings(suppressMessages(library(cowplot)))
```

```
ggplot(data=predicted.data, aes(x=rank, y=predicted.probability)) +
  geom_point(aes(color=real), alpha=1, shape=4, stroke=2) +
  xlab("Index") +
  ylab("Predicted probability of getting heart disease")
```



Por último, probamos el modelo para predecir el diagnóstico de una persona nueva. En este caso, de una mujer de 41 años.

```
prob <- predict(glmodel, data.frame(age=41, sex = "female", cp="atypical", trestbps=130, chol=204, fbs=
print(paste("La probabilidad de que padezca enfermedad cardiovascular es de", format(round(prob*100, 2)
```

```
## [1] "La probabilidad de que padezca enfermedad cardiovascular es de 97.88 %"
```

5. Conclusiones

Tras limpiar y analizar el conjunto de datos, podemos concluir que estos datos no son precisamente fáciles de relacionar, ya que aunque se presentan distintas variables, cada una tiene su peso y ninguna es definitiva a la hora de diagnosticar una enfermedad cardiovascular. Es por esto que sirve más para analizar el conjunto y generar modelos predictivos, más que encontrar causa-efecto entre las variables. Hemos visto que sí hay diferencia entre hombres y mujeres respecto a los resultados, así como que la frecuencia cardíaca máxima alcanzada está bastante relacionada con la edad. Hemos generado un modelo de regresión logística que nos ayuda a calcular la probabilidad de enfermedad teniendo en cuenta los valores que nos presentan las pruebas que han generado el conjunto de datos.

7. Resultado

Los datos analizados se pueden consultar en formato CSV en el repositorio, bajo el nombre `heart-clean.csv`.

```
#Exportación de datos en fichero CSV
write.csv(heartData, "heart-clean.csv")
```