Tipología y Ciclo de Vida de los Datos

Práctica 2 - Análisis de un dataset de enfermedades cardíacas.

Juan Luis Pérez Díez

ÍNDICE

Información del dataset

Introducción

Para esta segunda práctica de la asignatura de Tipología y Ciclo de Vida de los Datos hemos elegido analizar un dataset con información referente a enfermedades cardíacas que se puede obtener del "Machine Learning Repository" de la UCI [1].

En concreto, hemos focalizado nuestros esfuerzos en los datos recogidos en el "V.A. Medical Center, Long Beach and Cleveland Clinic Foundation" por el doctor Robert Detrano, por ser la más completa.

La fuente de datos contiene 14 variables de distintos tipos. Disponemos de atributos lógicos como la presencia de angina inducida por ejercicio, de medidas categóricas como el tipo de dolor que sufre el paciente en el pecho y de otras de tipo continuo como la presión sanguínea o los niveles de colesterol.

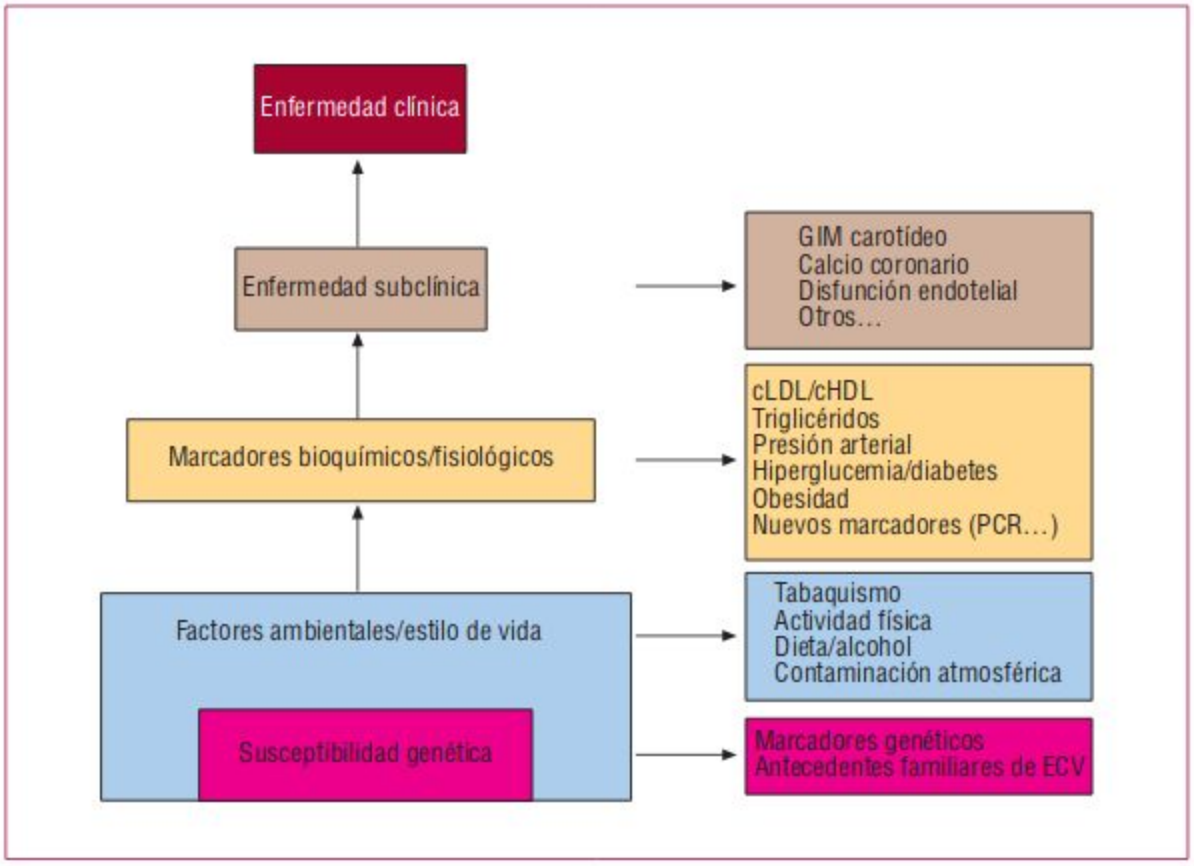
Con respecto al número de observaciones, las 302 que se recogen deberían ser suficientes para nuestros propósitos, ya que tenemos casos precedentes de su aplicación exitosa en tareas de predicción [2].

En relación al número de atributos y tamaño muestral creemos que puede ser una buena fuente a la hora de aplicar un amplio abanico de modelos de los utilizados en estadística y ciencia de datos.

Al disponer de una etiqueta que nos informa de si nos encontramos con un sujeto con enfermedad coronaria o no, trataremos de averiguar si el resto de variables nos permiten explicar y predecir la presencia o ausencia de tal enfermedad.

Idoneidad a la hora de obtener modelos útiles

Los factores de riesgo en la enfermedad cardiovascular son diversos y abarcan desde la susceptibilidad genética, los factores ambientales o el estilo de vida hasta otros como las características bioquímicas o fisiológicas [3].



En la literatura científica encontramos ejemplos de modelos de predicción. Alguno [4] usa para ello el nivel de colesterol, la presión arterial y las concentraciones de cLDL, pero sin duda el modelo más generalizado actualmente es el conocido como Test de Riesgo de Framingham [5] del que se ha demostrado su validez en grupos de población de varias regiones del mundo, incluyendo a los EEUU a los que se refieren nuestros datos [6].

El Test de Framingham tiene en cuenta las siguientes variables: edad, colesterol total, si se es fumador, nivel de cHDL y la presión sanguínea, dándole el mayor peso predictivo al factor de edad.

Un inconveniente es que en nuestro conjunto de datos carecemos de algunos atributos utilizados en los modelos que forman parte del estado del arte, como el

desglose del colesterol en función de si es cLDL o cHDL así como los datos referidos al consumo de tabaco. Esto nos imposibilita el uso de estos modelos directamente, pero esperamos ser capaces de suplir la carencia de unos datos con la presencia de otros muchos a nuestra disposición, de manera que seamos capaces de entrenar modelos que funcionen al menos igual de bien en cuanto a predicción.

Descripción del Dataset

Vamos a enumerar y describir los atributos de los que consta nuestra fuente según la información que nos da el proveedor:

1. **age**: La edad en años
2. **sex**: El sexo del sujeto (1 = hombre, 0 = mujer)
3. **cp**: El tipo de dolor en el pecho experimentado (1 = angina típica, 2 = angina atípica, 3 = dolor no asociado a angina, 4 = asintomático)
4. **trestbps**: Presión sanguínea en reposo medida en mmHg
5. **chol**: Valores de colesterol en mg/dl
6. **fbs**: Concentración de azúcar en sangre en ayunas (if > 120 mg/dl, 1 = si, 0 = no)
7. **restecg**: Medida del electrocardiograma en reposo (0 = normal, 1 = anomalía en la onda ST-T, 2 = probable hipertrofia en el ventrículo izquierdo según el criterio de Estes)
8. **thalach**: Máxima frecuencia cardíaca conseguida
9. **exang**: Angina inducida por ejercicio (1 = si, 0 = no)
10. **oldpeak**: Descenso de la ST inducida por el ejercicio en relación al reposo.
11. **slope**: Pendiente del segmento ST en el momento del pico de ejercicio (1= positiva, 2 = llana, 3 = negativa)
12. **ca**: Número de vasos sanguíneos principales coloreados por fluoroscopia (0-3)
13. **thal**: Enfermedad sanguínea conocida como Talasemia (3 = no, 6 = crónica, 7 = reversible)
14. **target**: Enfermedad coronaria (0 = no, 1-4 = si)

Ahora vamos a proceder a su análisis haciendo uso de RStudio. Para ello primeramente cargamos los datos en él y le asignamos nombres a las columnas.

```
> hdds <- read.csv("processed.cleveland.data")
> colnames(hdds) <- c("age", "sex", "cp", "trestbps",  "chol", "fbs",
"restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", "target")
```

Vamos a visualizar y describir los datos según se han cargado.

```
> str(hdds)
'data.frame':    302 obs. of  14 variables:
 $ age      : num  67 67 37 41 56 62 57 63 53 57 ...
 $ sex      : num  1 1 1 0 1 0 0 1 1 1 ...
 $ cp       : num  4 4 3 2 2 4 4 4 4 4 ...
 $ trestbps: num  160 120 130 130 120 140 120 130 140 140 ...
 $ chol     : num  286 229 250 204 236 268 354 254 203 192 ...
 $ fbs      : num  0 0 0 0 0 0 0 0 1 0 ...
```

```
$ restecg : num  2 2 0 2 0 2 0 2 2 0 ...
$ thalach : num  108 129 187 172 178 160 163 147 155 148 ...
$ exang   : num  1 1 0 0 0 0 1 0 1 0 ...
$ oldpeak : num  1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 0.4 ...
$ slope   : num  2 2 3 1 1 3 1 2 3 2 ...
$ ca      : Factor w/ 5 levels "?","0.0","1.0",..: 5 4 2 2 2 4 2 3 2 2 ...
$ thal    : Factor w/ 4 levels "?","3.0","6.0",..: 2 4 2 2 2 2 2 4 4 3 ...
$ target  : int  2 1 0 0 0 3 0 2 1 0 …
```

```
> head(hdds)
  age sex cp trestbps chol fbs restecg thalach exang oldpeak slope  ca thal target
1  67   1  4      160  286   0       2     108     1     1.5     2 3.0  3.0      2
2  67   1  4      120  229   0       2     129     1     2.6     2 2.0  7.0      1
3  37   1  3      130  250   0       0     187     0     3.5     3 0.0  3.0      0
4  41   0  2      130  204   0       2     172     0     1.4     1 0.0  3.0      0
5  56   1  2      120  236   0       0     178     0     0.8     1 0.0  3.0      0
6  62   0  4      140  268   0       2     160     0     3.6     3 2.0  3.0      3
```

```
> summary(hdds)
      age             sex               cp            trestbps         chol            fbs             restecg      
 Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :55.50   Median :1.0000   Median :3.000   Median :130.0   Median :241.5   Median :0.0000   Median :0.5000  
 Mean   :54.41   Mean   :0.6788   Mean   :3.166   Mean   :131.6   Mean   :246.7   Mean   :0.1457   Mean   :0.9868  
 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0   3rd Qu.:275.0   3rd Qu.:0.0000   3rd Qu.:2.0000  
 Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0   Max.   :564.0   Max.   :1.0000   Max.   :2.0000  
    thalach          exang          oldpeak          slope          ca          thal          target     
 Min.   : 71.0   Min.   :0.0000   Min.   :0.000   Min.   :1.000   ?  :  4   ?  :  2   Min.   :0.0000  
 1st Qu.:133.2   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   0.0:175   3.0:166   1st Qu.:0.0000  
 Median :153.0   Median :0.0000   Median :0.800   Median :2.000   1.0: 65   6.0: 17   Median :0.0000  
 Mean   :149.6   Mean   :0.3278   Mean   :1.035   Mean   :1.596   2.0: 38   7.0:117   Mean   :0.9404  
 3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:1.600   3rd Qu.:2.000   3.0: 20             3rd Qu.:2.0000  
 Max.   :202.0   Max.   :1.0000   Max.   :6.200   Max.   :3.000                       Max.   :4.0000  
```

Podemos observar que en las columnas *ca* y *thal* tenemos valores reflejados como '?' que trataremos en el apartado de limpieza y acondicionado de los datos.

Integración y selección

En este caso tenemos que la variable a estudiar es la etiqueta target. Además, sabemos que de los 76 atributos originales ya se han seleccionado previamente los 13 que mayor relación tienen con la presencia o ausencia de enfermedad cardíaca [1] y este último dataset es desde el que partiremos para realizar nuestros estudios.

Por todo esto hemos decidido no reducir el número de variables bajo estudio y dejar las 14 que lo integran.

La transformación que sí haremos es sobre la columna *target* porque, aunque se nos dan 5 niveles en la descripción de los datos, no se nos explica lo que representa cada uno, únicamente se nos dice que 0 es ausencia y 1-4 es presencia de enfermedad cardíaca. Este paso hemos decidido incluirlo dentro del tratamiento de valores nulos, que es cuando haremos la factorización del resto de variables.

Limpieza de los datos

Elementos nulos

Vamos a buscar elementos nulos en nuestro dataset. Para ello utilizaremos las siguientes funciones.

```
> colSums(is.na(hdds))
     age      sex       cp trestbps     chol      fbs  restecg  thalach    exang  oldpeak    slope       ca     thal   target 
       0        0        0        0        0        0        0        0        0        0        0        0        0        0 

> colSums(hdds == '?')
     age      sex       cp trestbps     chol      fbs  restecg  thalach    exang  oldpeak    slope       ca     thal   target 
       0        0        0        0        0        0        0        0        0        0        0        4        2        0 
```

Podemos ver que no tenemos ningún elemento nulo pero tenemos 6 ocurrencias de carácter “?” que debemos tratar.

En el caso de la columna *ca*, que corresponde con el número de vasos sanguíneos principales, hemos considerado adecuado sustituir las ocurrencias de ‘?’ por el valor más frecuente, que es 0 y que se repite con una frecuencia de más del doble que el segundo. Al haber tanta diferencia de frecuencias no esperamos introducir una gran cantidad de sesgo en la muestra haciendo esta suposición.

```
> hdds$ca[hdds$ca == "?"] = factor("0.0")
> hdds$ca <- droplevels(hdds$ca)
```

Para el caso de *thal* el segundo valor más frecuente está a 50 ocurrencias del primero. Dividiendo frecuencia entre el total de filas tenemos un 55% de ocurrencias del más frecuente y un 39% del segundo. Por ello no nos parece adecuado sustituir el valor por el más frecuente ya que podríamos sesgar demasiado los datos, con lo que ello supone para las aplicaciones médicas.

En su lugar, vamos a imputar los valores mediante el uso de los k-nearest-neighbours incluído en la librería *VIM*.

```
> install.packages("VIM")
> library(VIM)
> hdds[hdds$thal == '?',]
      age sex cp trestbps chol fbs restecg thalach exang oldpeak slope  ca thal
target
87    53   0  3      128  216   0       2     115   0       0     1 0.0  ?      0
266   52   1  4      128  204   1       0     156   1       1     2 0.0  ?      2
```

Convertimos las ocurrencias de "?" en *NA*, aplicamos el kNN y comprobamos el resultado.

```
> hdds$thal[hdds$thal == "?"] = NA
> colSums(is.na(hdds))
age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
0     0     0     0     0     0     0     0     0     0     0     0     2
0
> hdds$thal <- kNN(hdds)$thal
> colSums(is.na(hdds))
age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
0     0     0     0     0     0     0     0     0     0     0     0     0
0
> hdds$thal <- droplevels(hdds$thal)
```

En ambos casos hemos aprovechado para eliminar los niveles de factor que han quedado vacíos.

Ahora vamos a aprovechar y factorizar todas las columnas que sabemos que contienen medidas categóricas.

```
> for (i in c("sex", "cp", "fbs", "restecg", "exang", "slope", "target")) {
+     hdds[,i] <- as.factor(hdds[,i])
+     }
> remove(i)
```

Además vamos a cambiar el nombre de los niveles para que nos sea más fácil la interpretación de los datos a partir de ahora.

```
> levels(hdds$sex) <- c("f", "m")
> levels(hdds$cp) <- c("typical", "atypical", "non-anginal", "asymptomatic")
> levels(hdds$restecg) <- c("normal", "abnormal", "hypertrophy")
> levels(hdds$slope) <- c("upsloping", "flat", "downsloping")
> levels(hdds$thal) <- c("no", "chronic", "reversible")
```

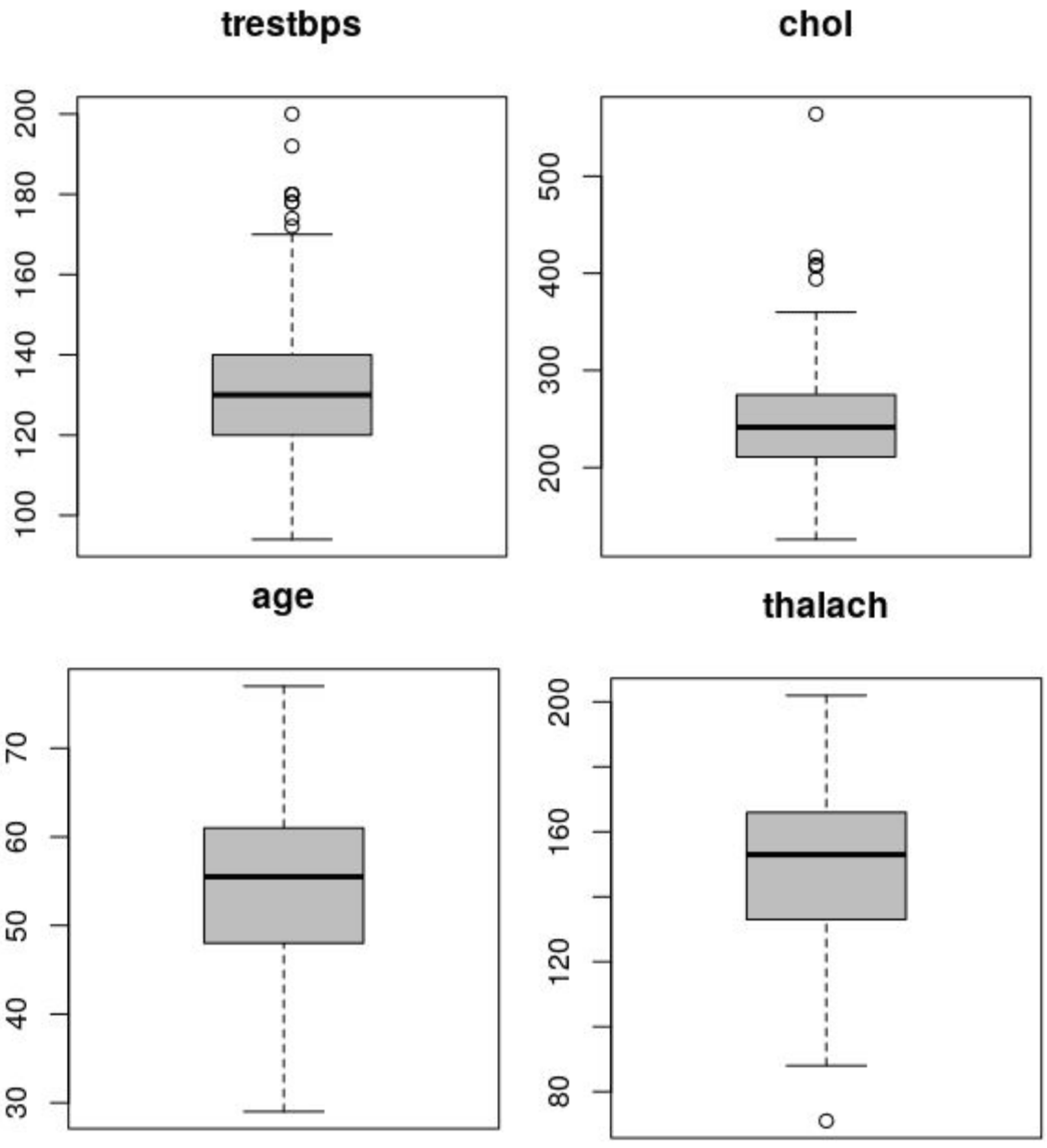
Por último vamos a transformar la columna *target* como hemos indicado antes de manera que pasará a representar una variable binaria con la ausencia o presencia de enfermedad cardíaca.

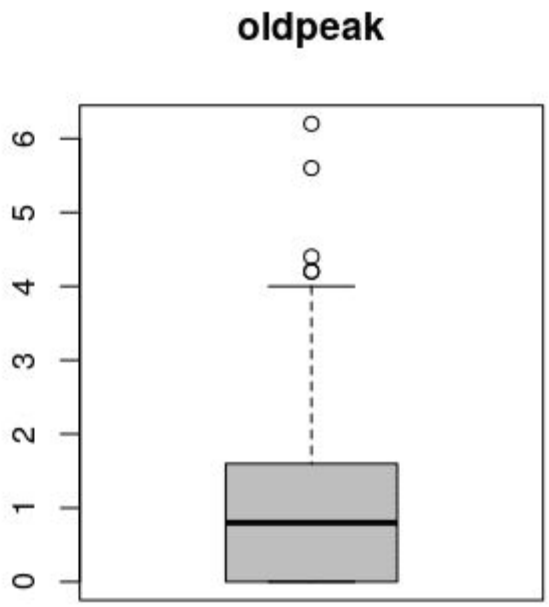
```
> hdds$target[hdds$target != 0] = factor("1")
> hdds$target <- droplevels(hdds$target)
```

Valores extremos

A la hora de estudiar posibles anomalías u outliers en los datos vamos a realizar primeramente una exploración visual por medio de boxplots aplicados a nuestras 5 variables numéricas, que sabemos bastante útil.

```
> boxplot(hdds$age, ylab="valor", main="age", col="grey")
> boxplot(hdds$trestbps, ylab="valor", main="trestbps", col="grey")
> boxplot(hdds$chol, ylab="valor", main="chol", col="grey")
> boxplot(hdds$thalach, ylab="valor", main="thalach", col="grey")
> boxplot(hdds$oldpeak, ylab="valor", main="oldpeak", col="grey")
```





Menos para el atributo edad parece que podemos distinguir visualmente varios outliers.

Para estudiarlos en más detalle y poder detectarlos numéricamente vamos a utilizar la función *boxplot.stats* [7]. A la hora de elegir el coeficiente que determina cuánto queremos extender los bigotes vamos a hacer pruebas con varios valores (2, 2.5 y 3).

```
> for (c in list(2,2.5,3)) {
+     message("Coef=",c)
+     for (i in c("trestbps", "chol", "thalach", "oldpeak")) {
+     x <- boxplot.stats(hdds[,i], coef=c)
+     message(i)
+     print(x$stats)
+     print(x$out)
+     }
+ }
```

Para cada atributo nos vamos a quedar con dos de sus estadísticas. La primera, *stats*, nos devuelve el extremo del bigote menor, el valor menor donde empieza la caja, la mediana, el valor mayor donde termina la caja y el extremo superior del bigote, por ese orden. La segunda, *out*, nos devuelve los valores que quedan fuera del rango especificado.

```
Coef=2
trestbps
[1]  94 120 130 140 180
[1] 200 192
chol
```

```
[1] 126.0 211.0 241.5 275.0 394.0
[1] 417 407 564 409
thalach
[1]  71 133 153 166 202
numeric(0)
oldpeak
[1] 0.0 0.0 0.8 1.6 4.4
[1] 6.2 5.6
Coef=2.5
trestbps
[1]  94 120 130 140 180
[1] 200 192
chol
[1] 126.0 211.0 241.5 275.0 417.0
[1] 564
thalach
[1]  71 133 153 166 202
numeric(0)
oldpeak
[1] 0.0 0.0 0.8 1.6 5.6
[1] 6.2
Coef=3
trestbps
[1]  94 120 130 140 200
numeric(0)
chol
[1] 126.0 211.0 241.5 275.0 417.0
[1] 564
thalach
[1]  71 133 153 166 202
numeric(0)
oldpeak
[1] 0.0 0.0 0.8 1.6 6.2
numeric(0)
```

A la vista de los resultados vamos a intentar razonar qué valores no nos parecen "naturales".

- Para *trestbps*, la presión sanguínea, valores de hasta 200 no parecen fuera de lo común en pacientes de este tipo. Son valores alarmantes pero compatibles con la vida [8] y además para ambos casos sabemos que el paciente fue diagnosticado con enfermedad coronaria. Por ello vamos a dejarlos sin modificar.

- En el atributo *chol*, el colesterol total, tenemos un valor, 564, muy fuera de rango. Está unos 150 puntos por encima del siguiente, 417, y corresponde a una persona sin enfermedad. Vamos a descartar esta observación completamente y dejar el resto como están.

```
> hdds <- hdds[!(hdds$chol == 564),]
> summary(hdds$chol)
   Min. 1st Qu.  Median  Mean 3rd Qu. Max.
  126.0   211.0   241.0   245.7   275.0   417.0
```

- Para *thalach*, máxima frecuencia cardiaca conseguida, no nos salen datos significativos con ningún coeficiente de los probados y además podemos ver que el valor inferior del boxplot se corresponde con una persona enferma así que ese valor podría tener sentido.

- En el caso de *oldpeak*, el descenso de la ST inducida por el ejercicio en relación al reposo, tenemos uno o dos valores llamativos. En este caso el problema que se nos presenta es la falta de conocimiento en el dominio sobre el que tratamos, y tras una búsqueda rápida por PubMed no encontramos nada definitivo sobre los valores aceptables. Tratándose de personas diagnosticadas y como el resto de decisiones que hemos tomado se corresponden con los valores que resultan de aplicar un coeficiente de 3, vamos a hacer caso a estos resultados y dejar los valores tal como están.

Exportación de los datos limpios

Ahora que tenemos el dataset limpito vamos a proceder a su volcado en otro fichero *.csv* para tenerlo a mano a partir de ahora.

```
> write.csv(hdds, "cleaned.cleveland.csv")
```

Análisis de los datos

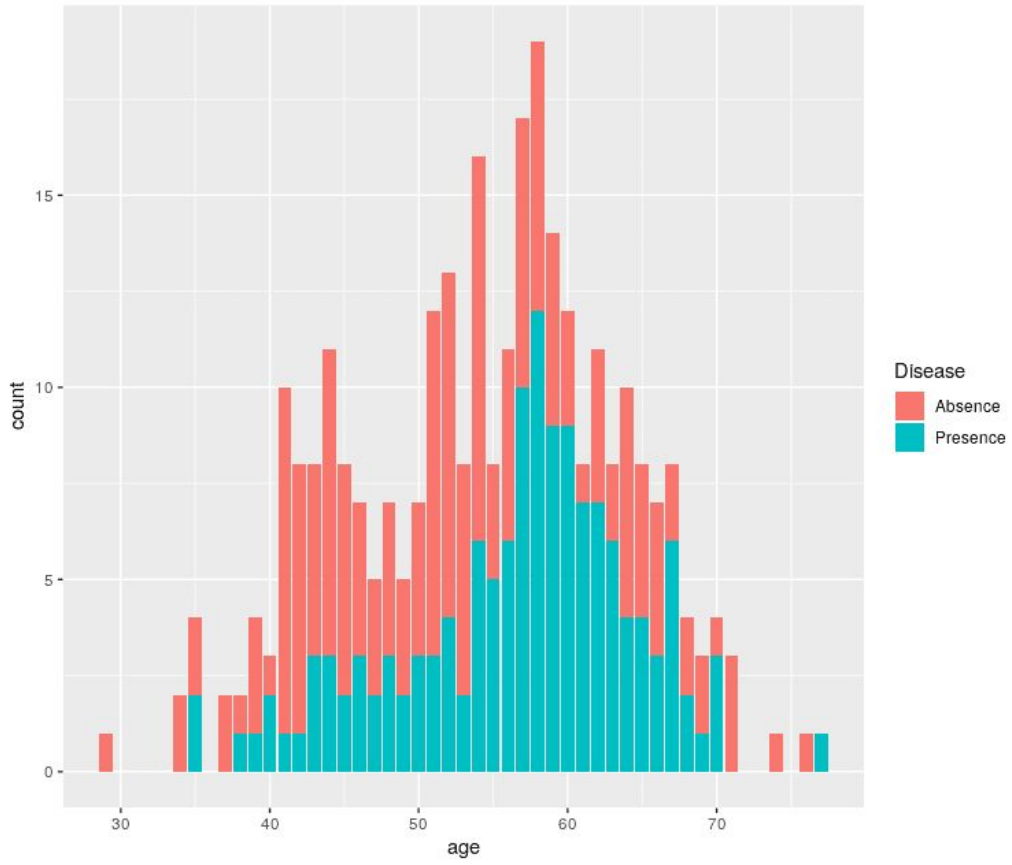
Distribuciones

Lo primero que vamos a hacer de cara al análisis es un graficado de las variables, de manera que podamos hacernos una idea de cómo se distribuyen en función de la presencia o ausencia de enfermedad coronaria.

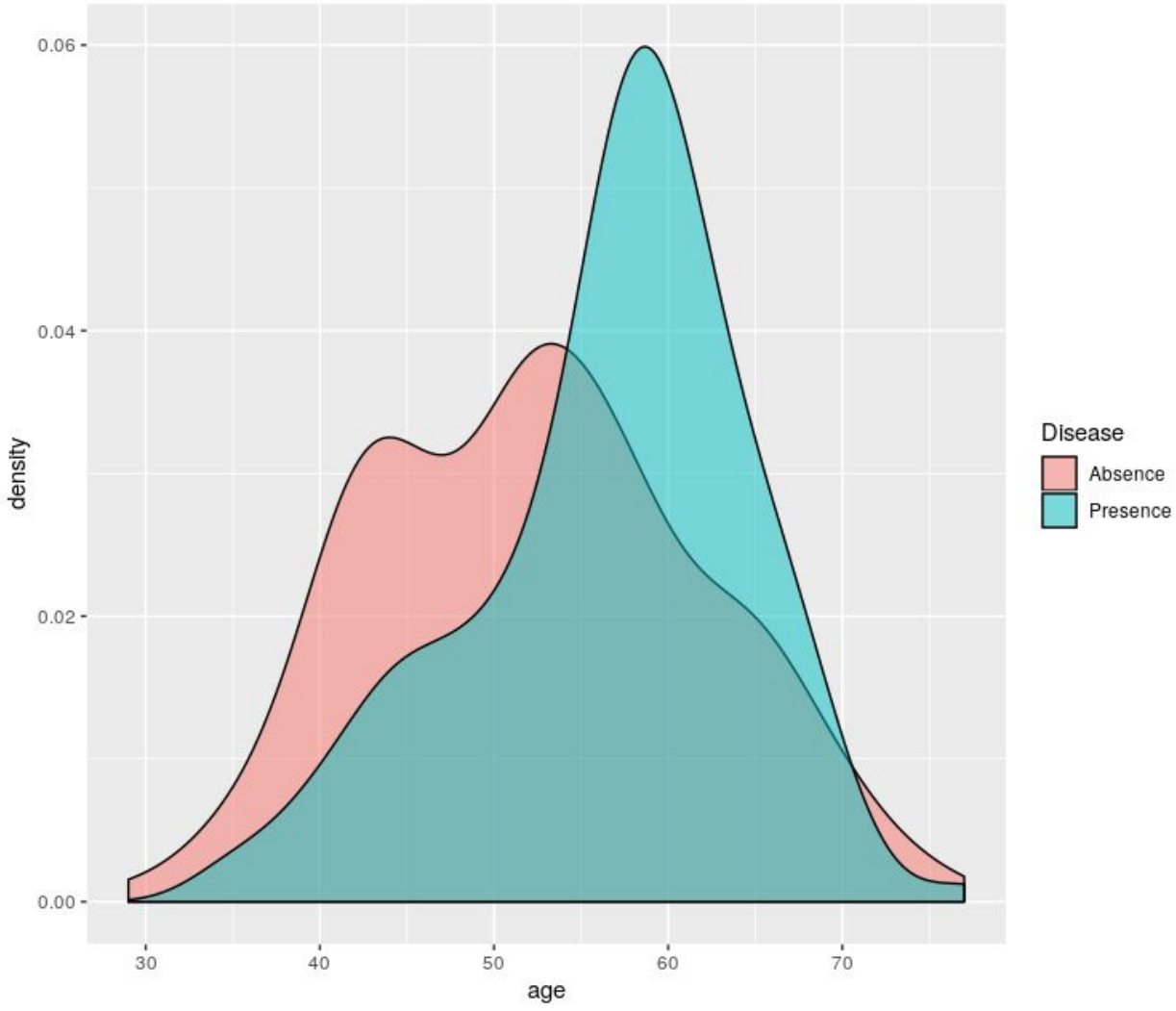
Para cada atributo vamos a generar dos gráficas. La primera de ellas será un diagrama de frecuencia. La segunda nos permitirá ver mejor el impacto de cada una sobre la enfermedad coronaria, para ello usaremos diagramas de frecuencia "llenos" en los atributos discretos y curvas de densidad en los continuos.

Age

```
> library(ggplot2)
> ggplot(data=hdds,aes(x=age,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

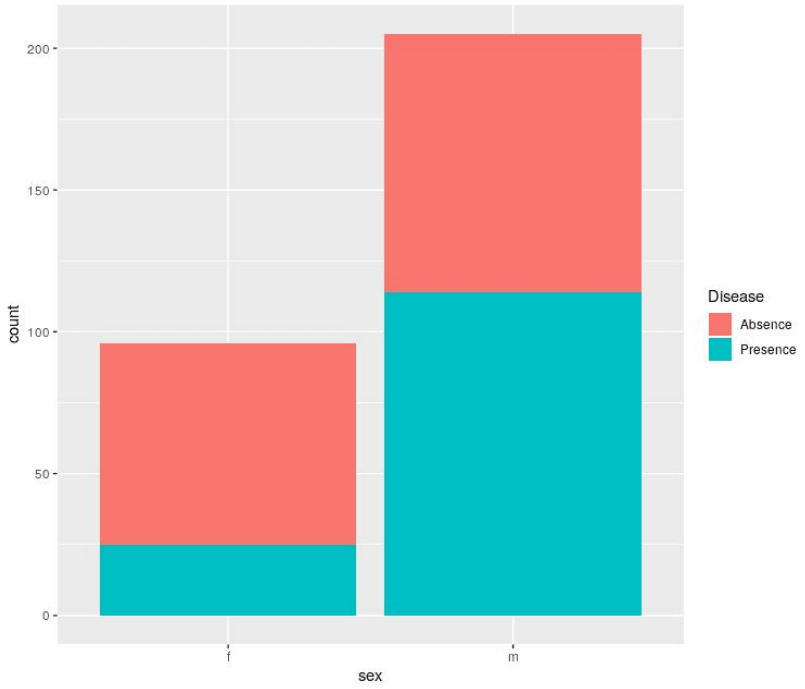


```
> ggplot(data=hdds,aes(x=age,fill=target)) + geom_density(alpha=0.5) +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

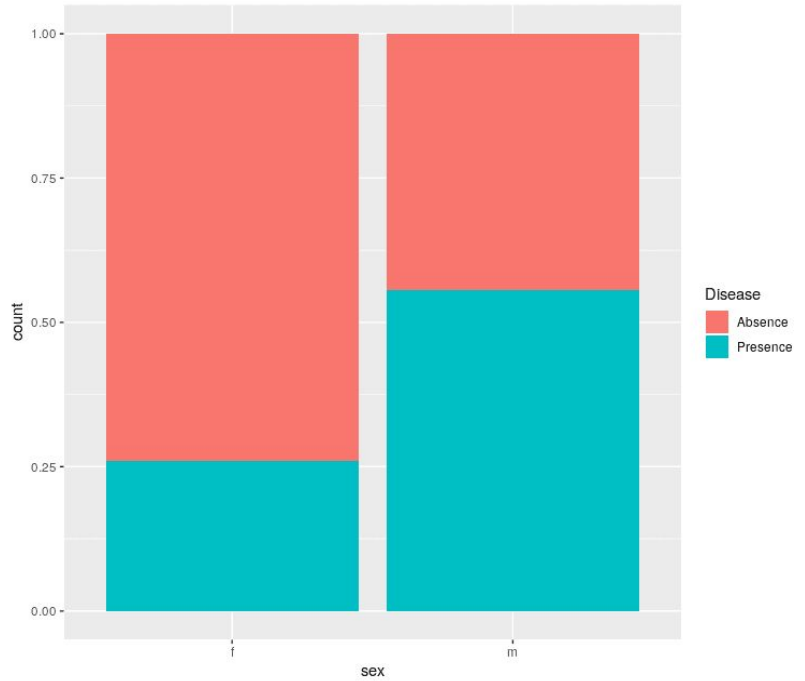


Sex

```
> ggplot(data=hdds,aes(x=sex,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

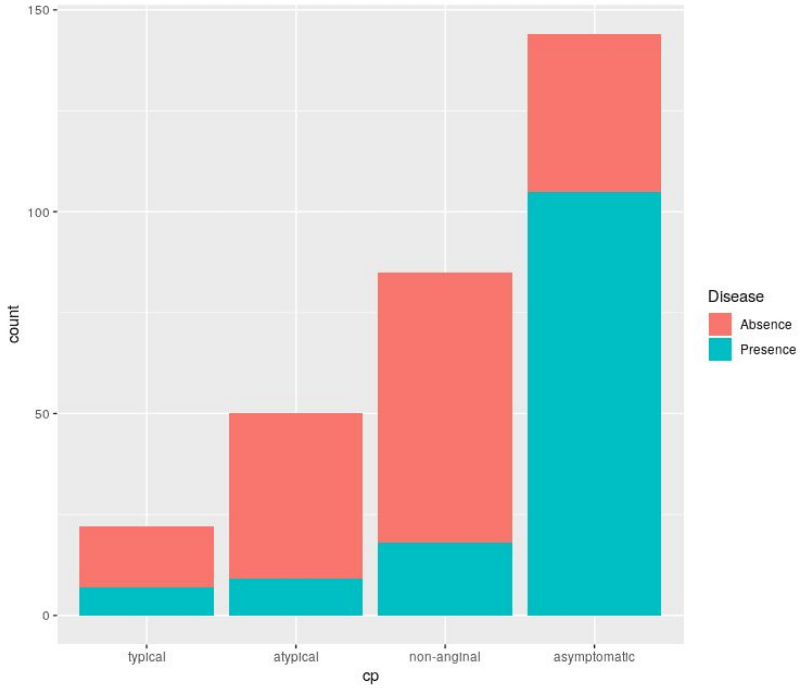



```
> ggplot(data=hdds,aes(x=sex,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

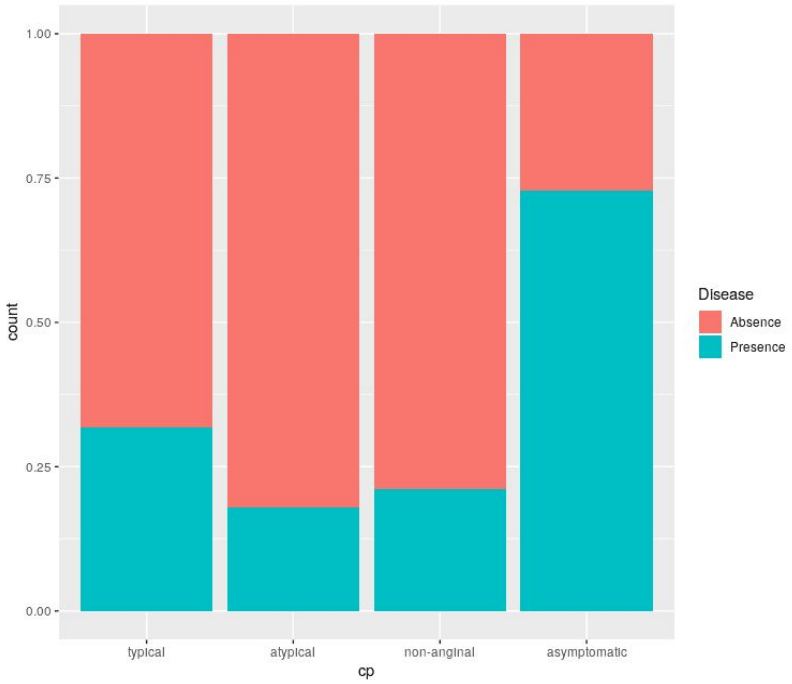


Cp

```
> ggplot(data=hdds,aes(x=cp,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

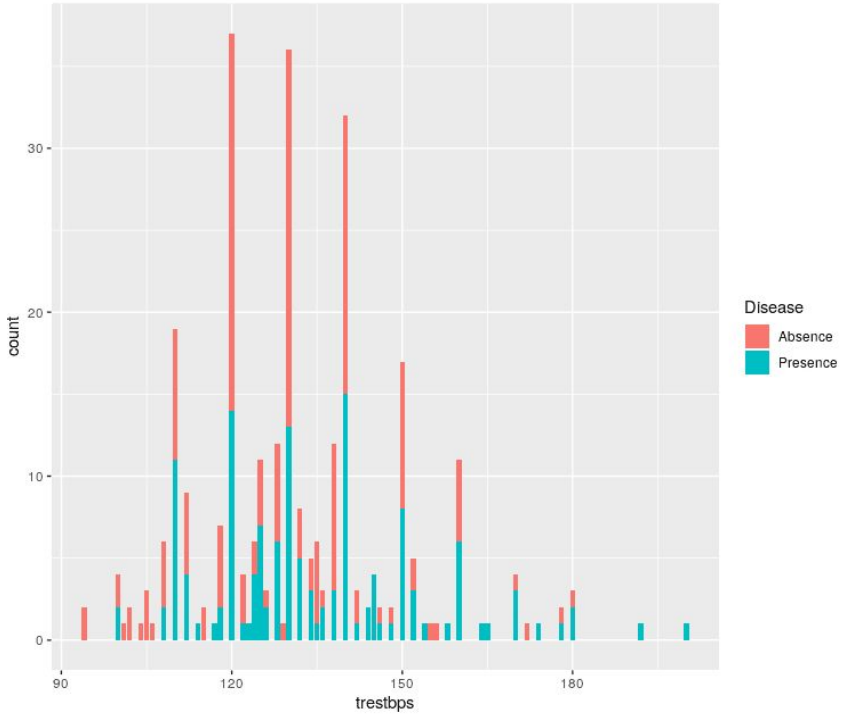



```
> ggplot(data=hdds,aes(x=cp,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

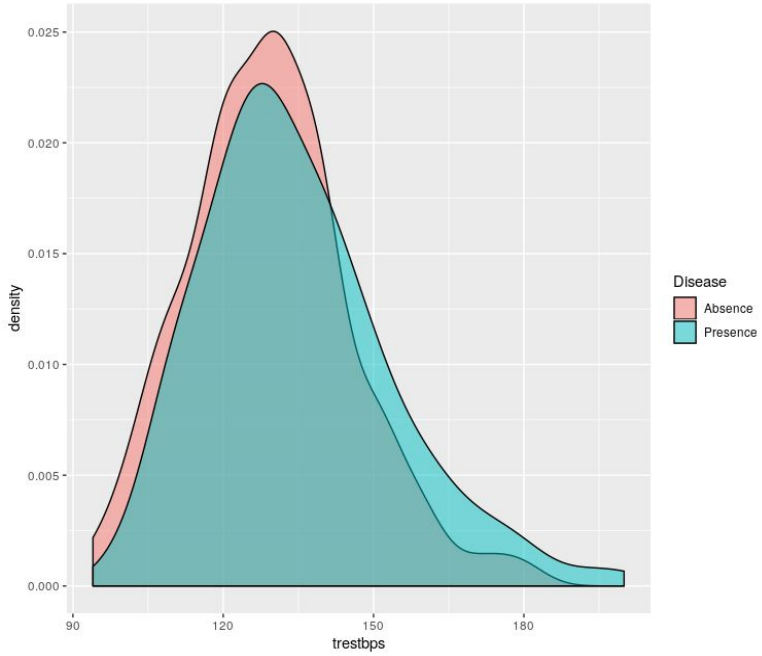


Trestbps

```
> ggplot(data=hdds,aes(x=trestbps,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

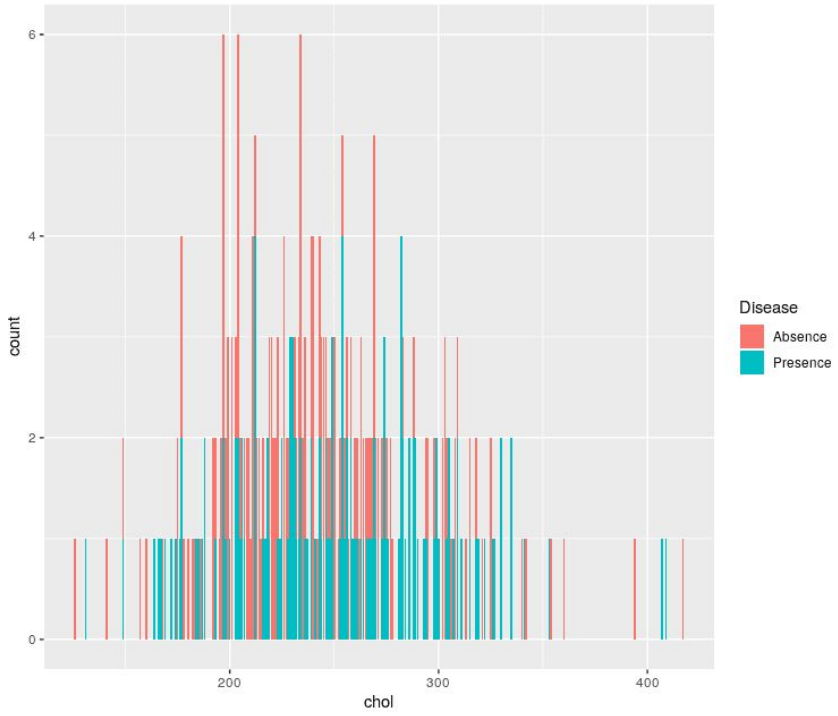



```
> ggplot(data=hdds,aes(x=trestbps,fill=target)) + geom_density(alpha=0.5) +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

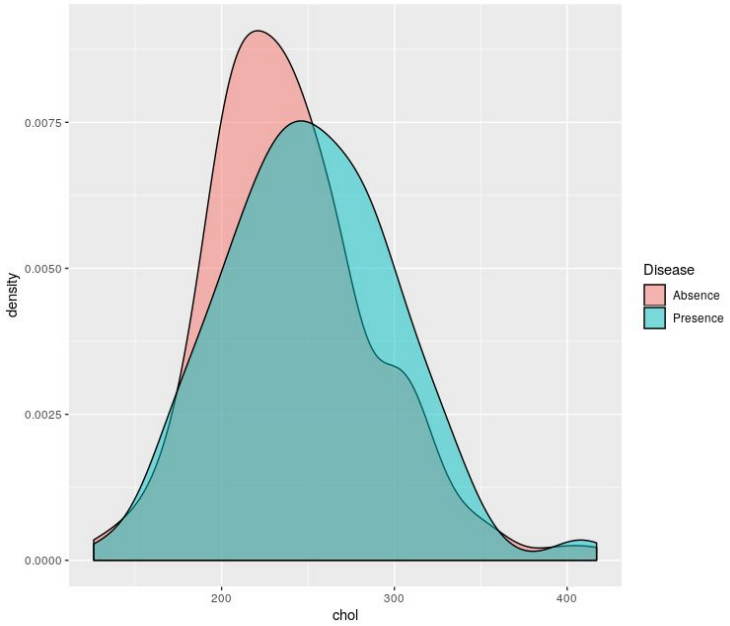


Chol

```
> ggplot(data=hdds,aes(x=chol,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

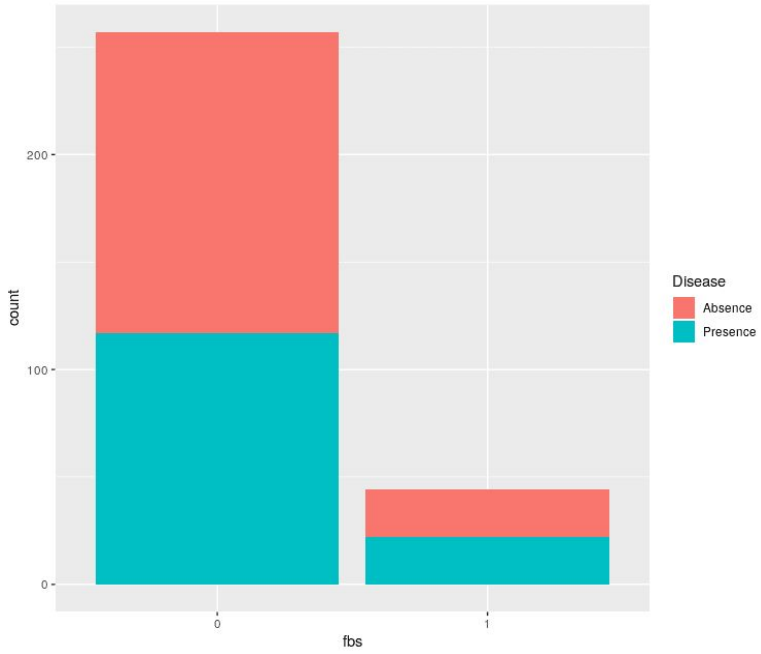



```
> ggplot(data=hdds,aes(x=chol,fill=target)) + geom_density(alpha=0.5) +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

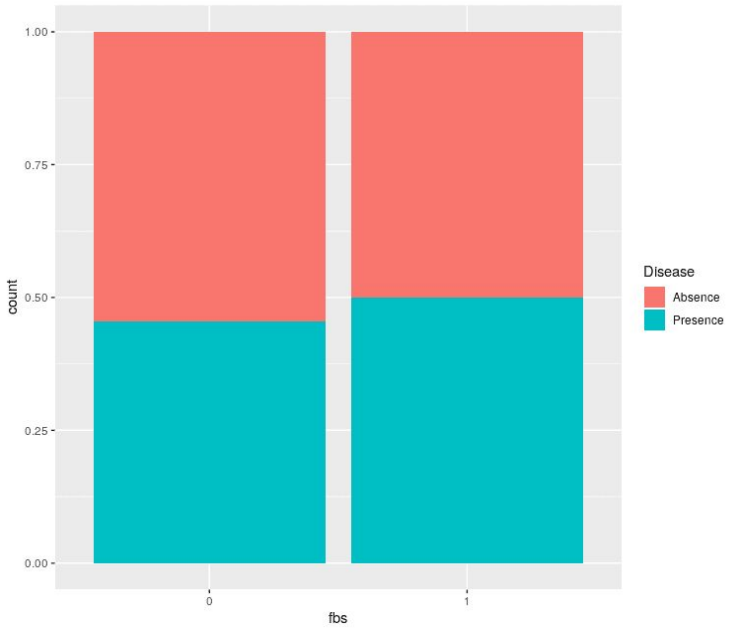


Fbs

```
> ggplot(data=hdds,aes(x=fbs,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

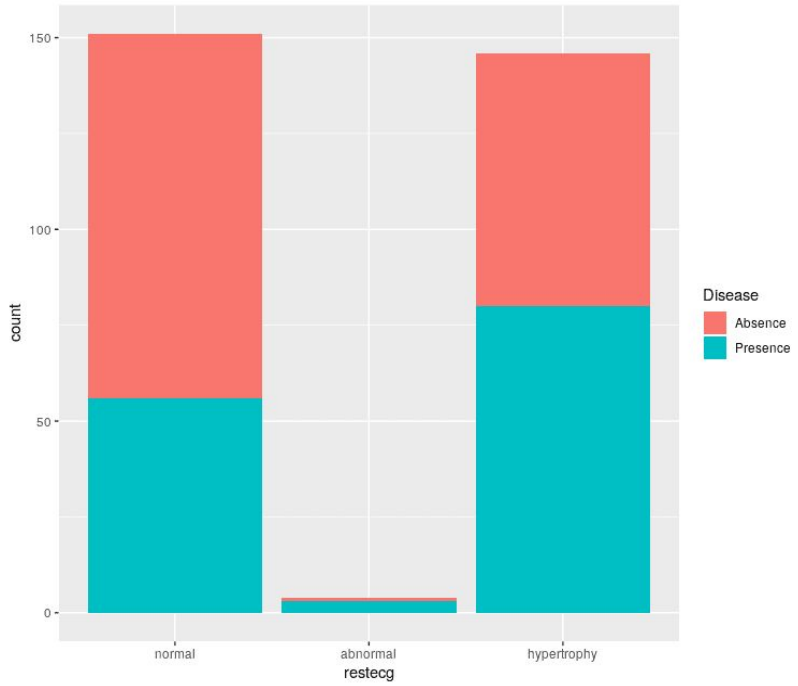



```
> ggplot(data=hdds,aes(x=fbs,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

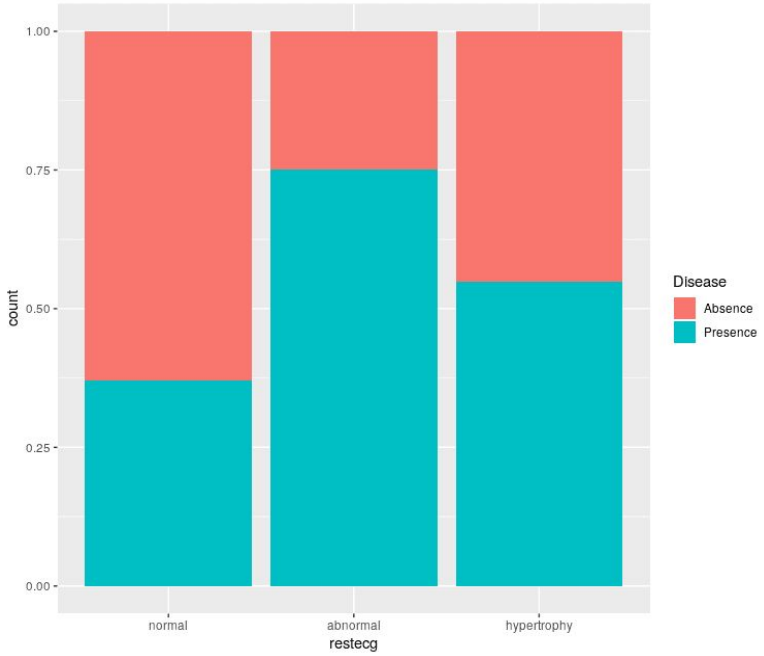


Restecg

```
> ggplot(data=hdds,aes(x=restecg,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

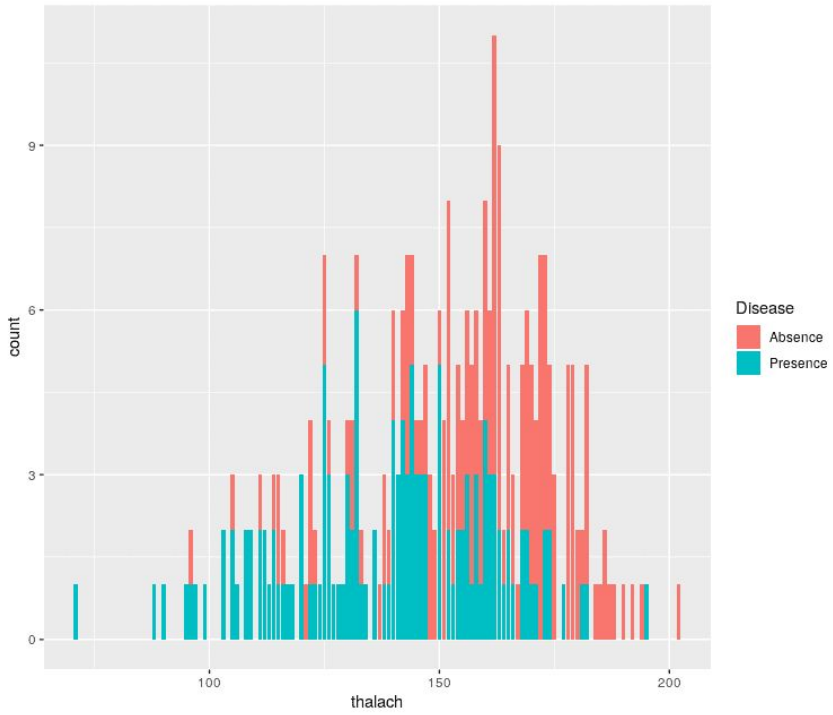



```
> ggplot(data=hdds,aes(x=restecg,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

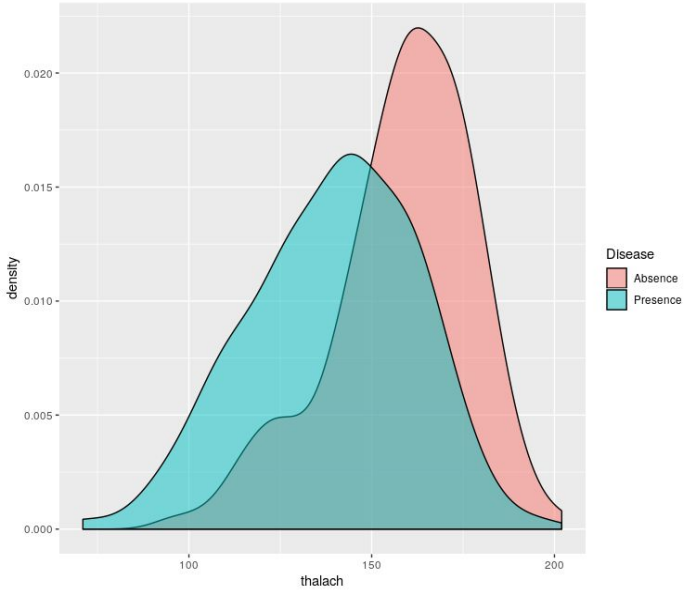


Thalach

```
> ggplot(data=hdds,aes(x=thalach,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

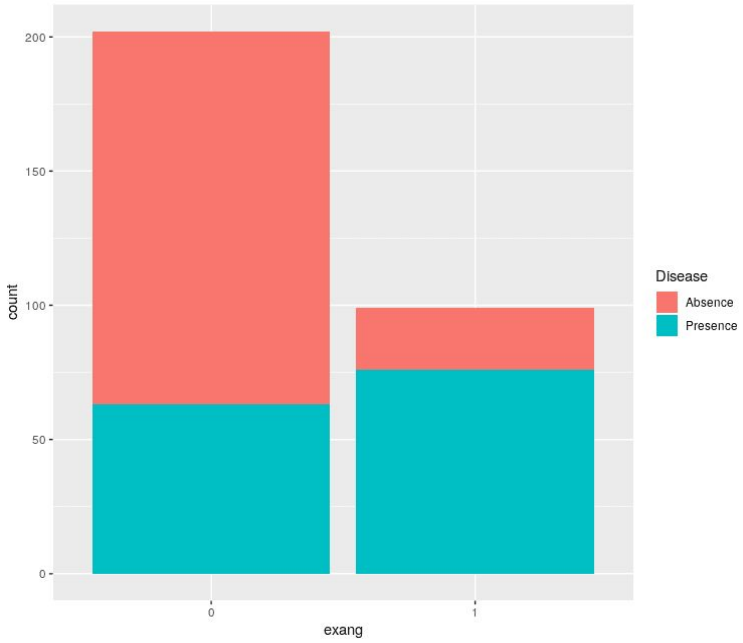



```
> ggplot(data=hdds,aes(x=thalach,fill=target)) + geom_density(alpha=0.5) +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

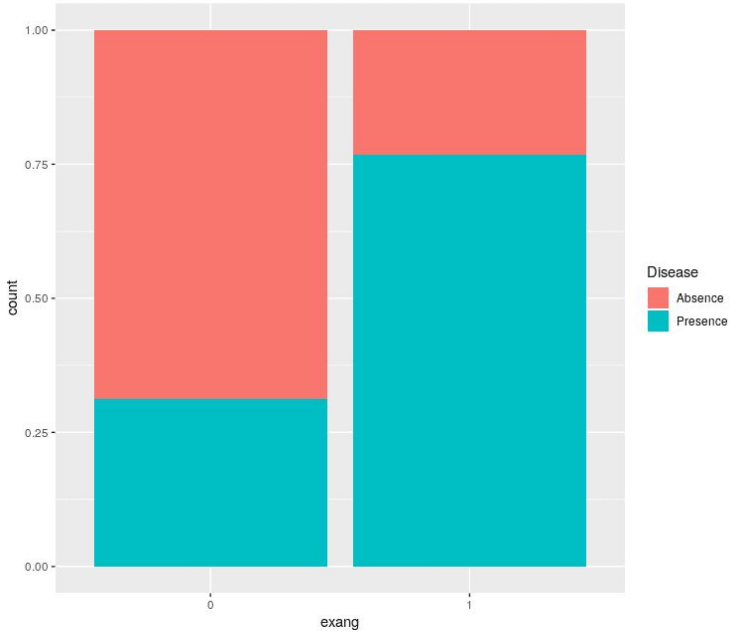


Exang

```
> ggplot(data=hdds,aes(x=exang,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

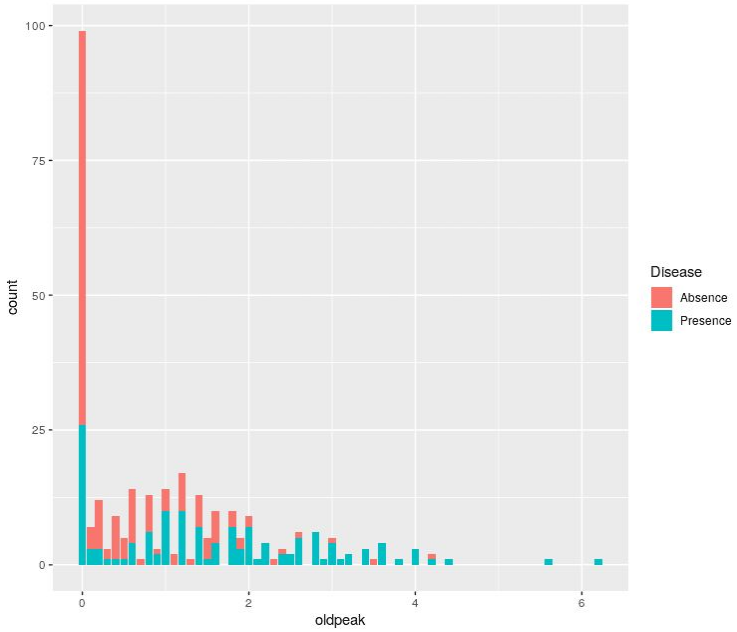



```
> ggplot(data=hdds,aes(x=exang,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

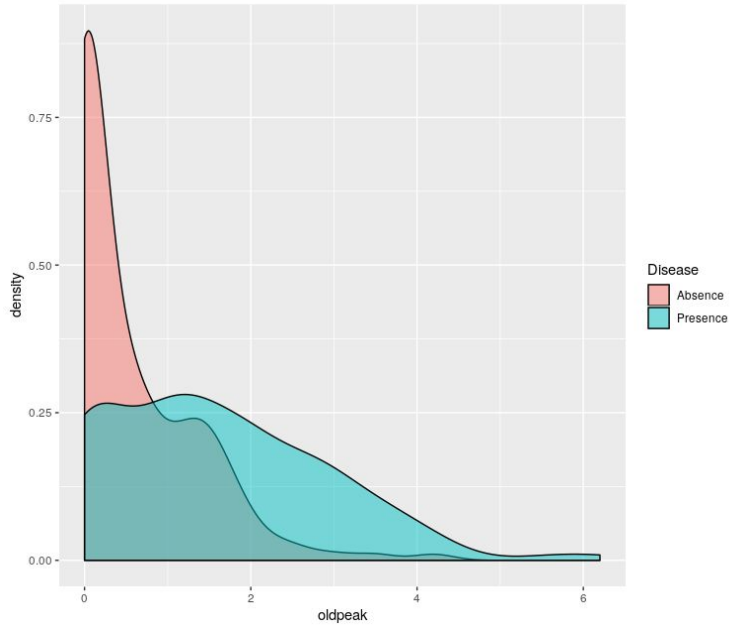


Oldpeak

```
> ggplot(data=hdds,aes(x=oldpeak,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

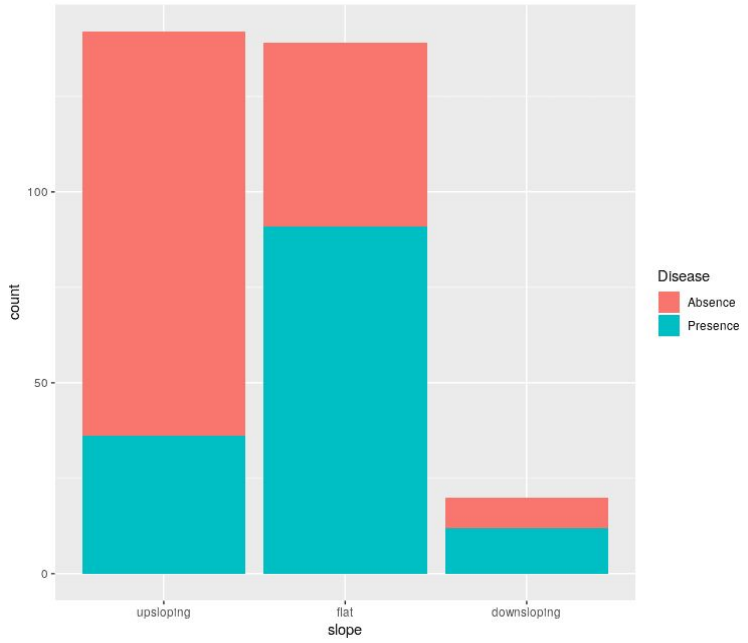



```
> ggplot(data=hdds,aes(x=oldpeak,fill=target)) + geom_density(alpha=0.5) +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

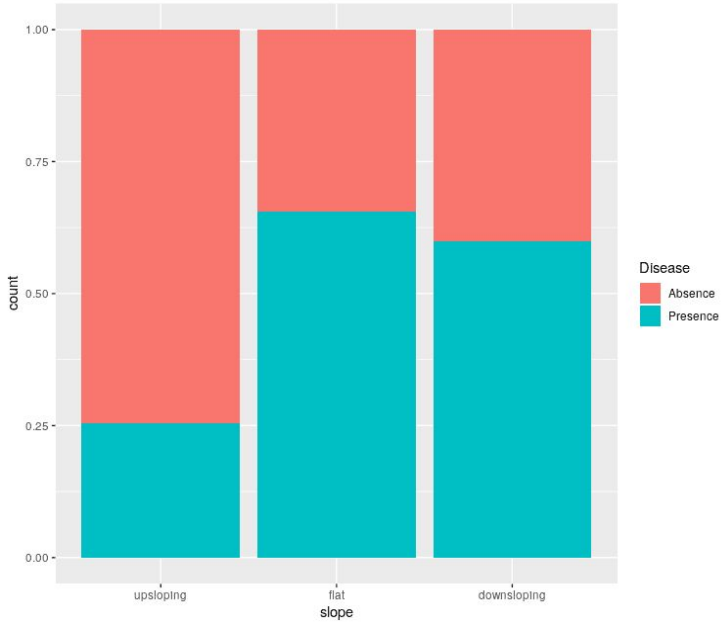


Slope

```
> ggplot(data=hdds,aes(x=slope,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

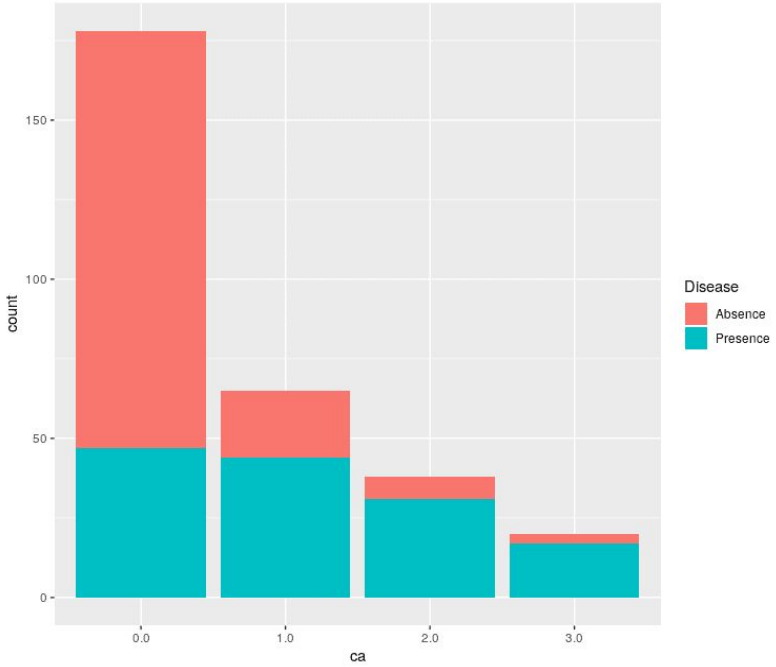



```
> ggplot(data=hdds,aes(x=slope,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

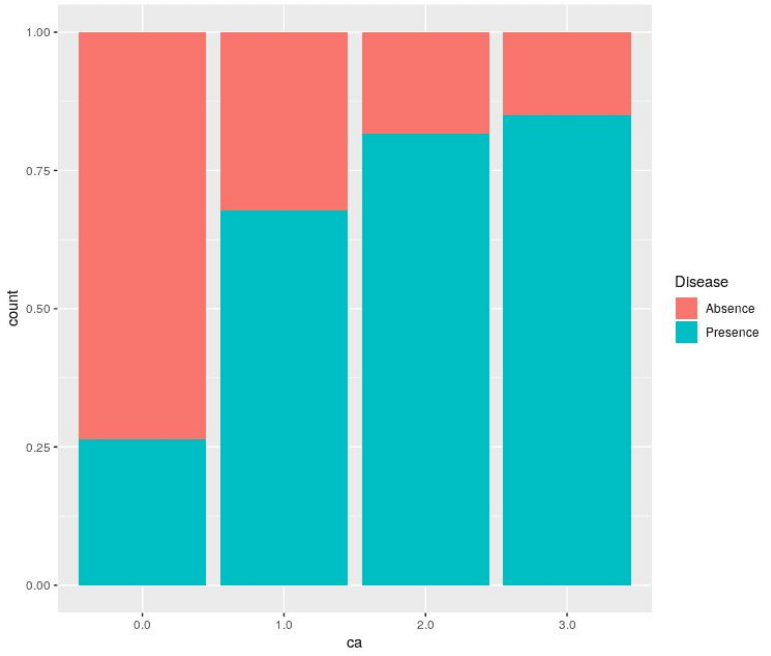


Ca

```
> ggplot(data=hdds,aes(x=ca,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

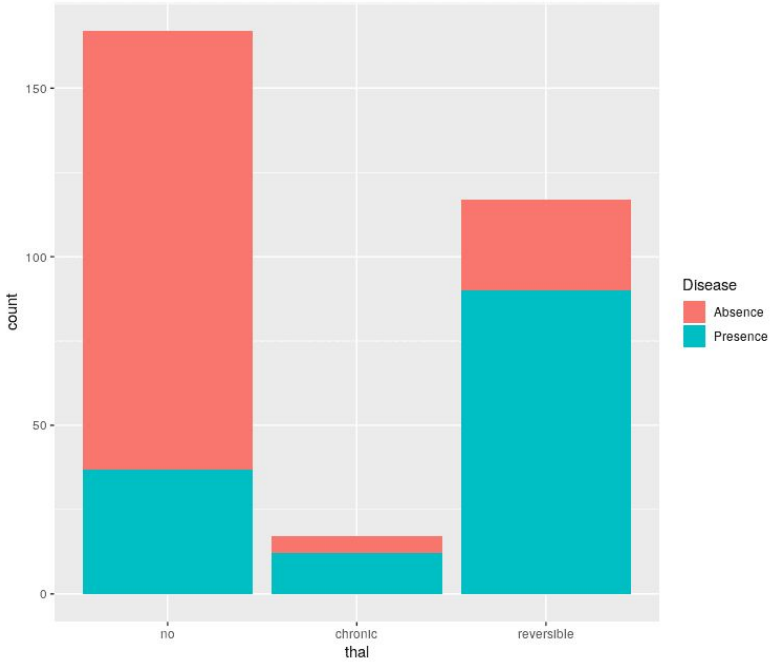



```
> ggplot(data=hdds,aes(x=ca,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

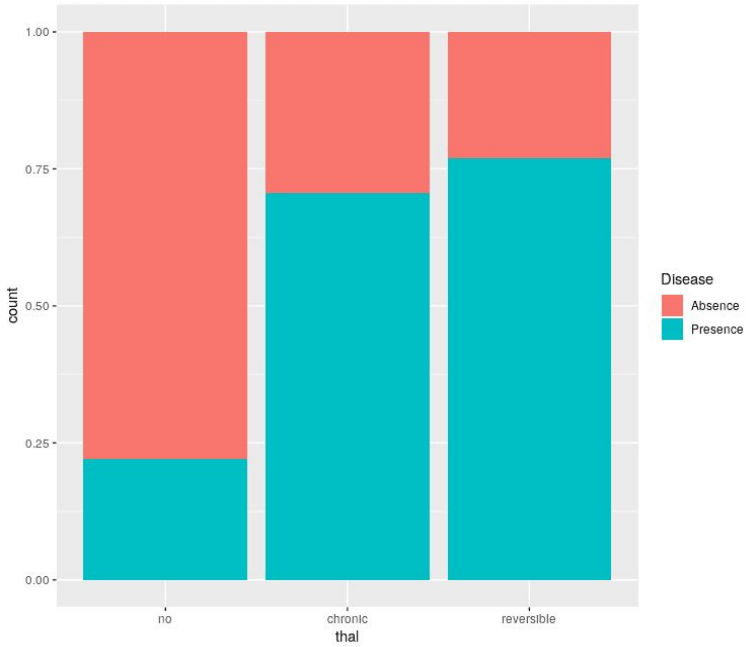


Thal

```
> ggplot(data=hdds,aes(x=thal,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```

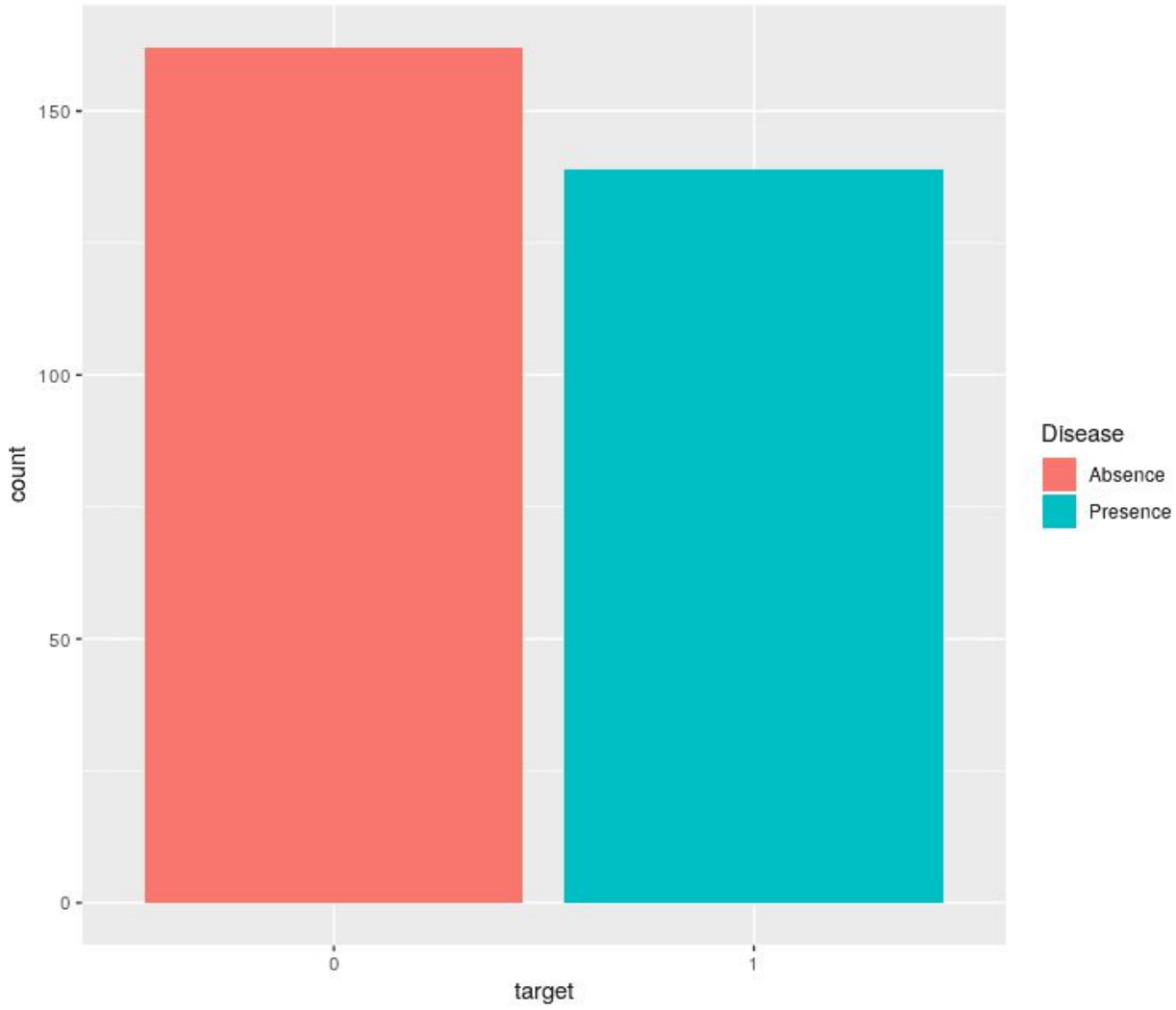



```
> ggplot(data=hdds,aes(x=thal,fill=target)) + geom_bar(position="fill") +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```



Target

```
> ggplot(data=hdds,aes(x=target,fill=target)) + geom_bar() +
scale_fill_discrete(name="Disease",labels=c("Absence","Presence"))
```



Planificación de los análisis

Hemos propuesto tres tipos de pruebas estadísticas a realizar: análisis de correlación, contraste de hipótesis y regresión logística.

De cara a la correlación haremos dos estudios enfocados a ver cuales son las variables que más influyen a la hora de predecir la presencia o ausencia de enfermedad cardíaca. El primer análisis lo efectuaremos únicamente utilizando variables cuantitativas mientras que para el segundo utilizaremos todas, las cuantitativas y las cualitativas. De esta manera podremos saber qué variables de nuestro dataset son las que más influyen a la hora de predecir la presencia de enfermedad coronaria.

Para el contraste de hipótesis haremos una única prueba, para ver si podemos asegurar que el colesterol es mayor en pacientes que presentan enfermedad frente a los que no la presentan, que es algo a lo que se le suele dar mucha importancia en ambientes generalistas pero que sospechamos no es tan determinante como se suele pensar.

De cara a la regresión logística construiremos varios modelos seleccionando distintas variables. Empezaremos con las variables que tenemos y que se corresponden con el Test de Riesgo de Framingham [5] para luego pasar a otros conjuntos que consideremos que pueden mejorar el modelo con las variables que tenemos que más correlacionan.

Comprobación de la normalidad y homogeneidad

Normalidad

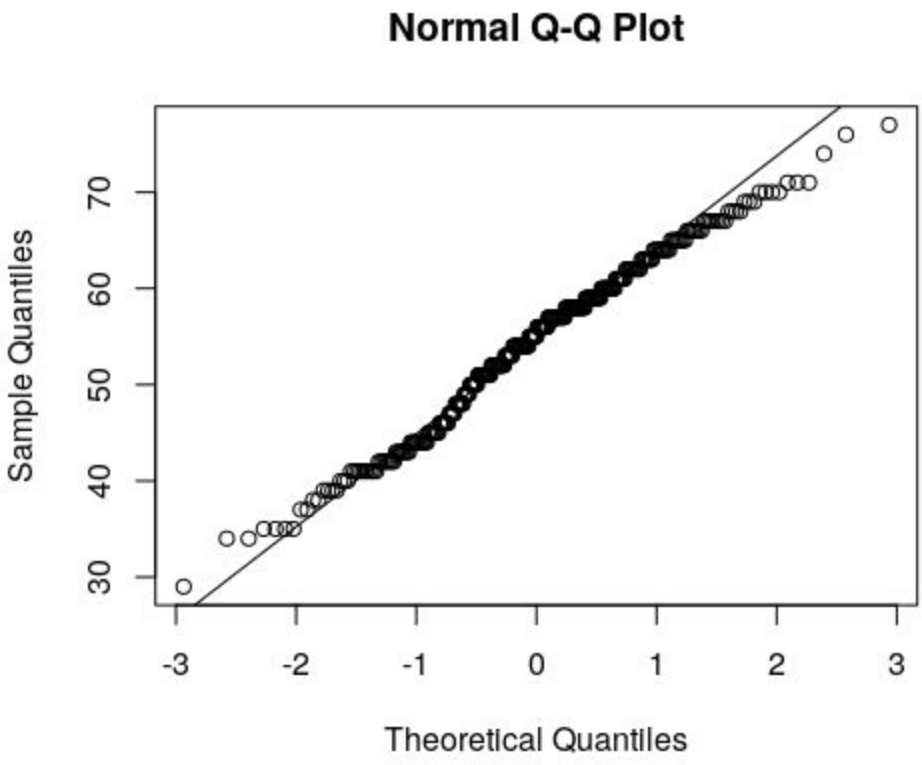
Tenemos varias maneras de comprobar la normalidad. Podemos aplicar el teorema del límite central, se puede realizar una visualización gráfica con las curvas Q-Q o aplicar tests como los de Shapiro-Wilk o el de Anderson-Darling.

Según el teorema del límite central, al tener una cantidad de muestras grande y superior a 30 podremos asumir normalidad para todas las variables, en cualquier caso vamos a aplicar más métodos.

Visualmente, a parte de ver las gráficas de densidad como ya hemos hecho, podemos hacer uso de las curvas QQ. Vamos a generarlas para los atributos cuantitativos. Además, haremos el test de Shapiro-Wilk para cada una de las variables.

Age

```
> qqnorm(hdds$age)
> qqline(hdds$age)
```

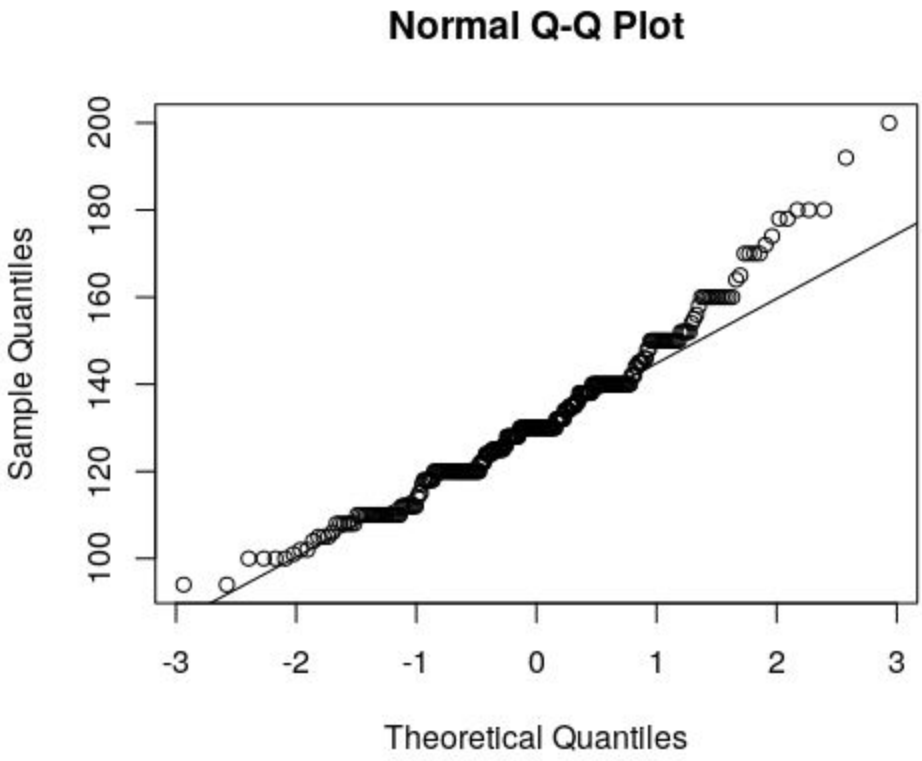


```
> shapiro.test(hdds$age)
Shapiro-Wilk normality test
data:  hdds$age
W = 0.98697, p-value = 0.008089
```

Sabemos que W va de 0 a 1 y de menor a mayor normalidad. A la vista de la gráfica y del valor de W podemos decir que la edad sigue una distribución normal.

Trestbps

```
> qqnorm(hdds$trestbps)
> qqline(hdds$trestbps)
```

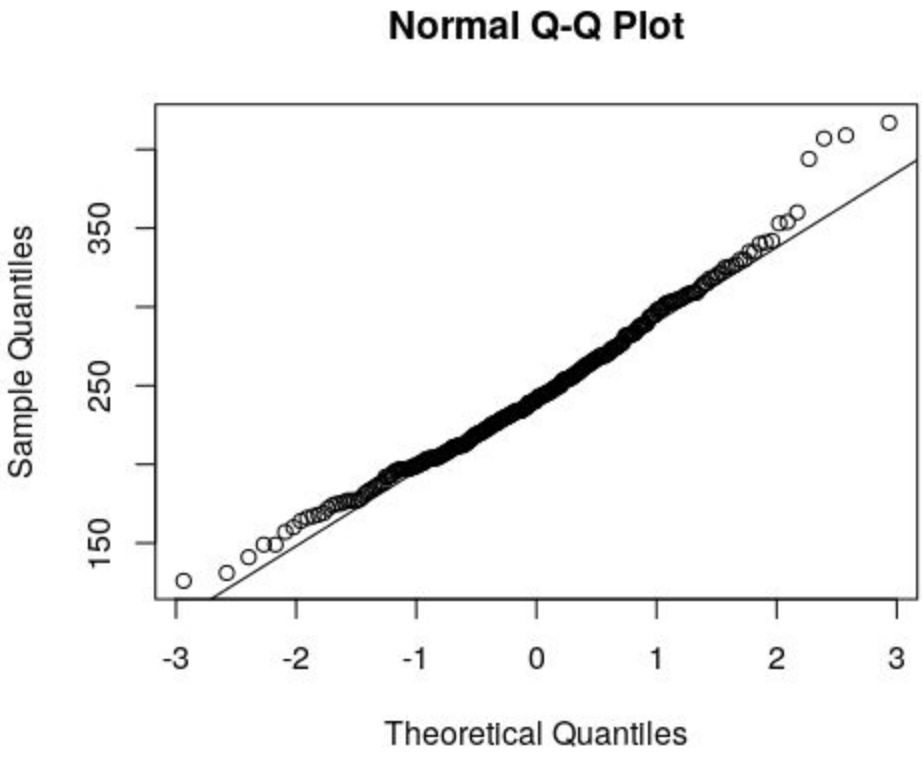



```
> shapiro.test(hdds$trestbps)
Shapiro-Wilk normality test
data:  hdds$trestbps
W = 0.96611, p-value = 1.673e-06
```

También podemos asumir normalidad.

Chol

```
> qqnorm(hdds$chol)
> qqline(hdds$chol)
```

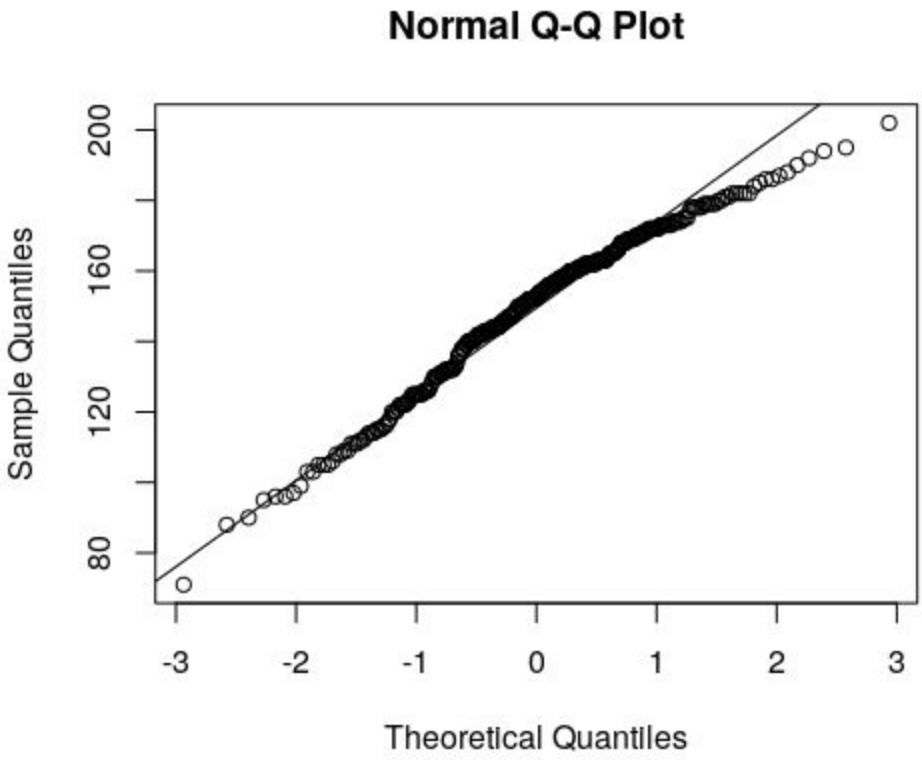



```
> shapiro.test(hdds$trestbps)
Shapiro-Wilk normality test
data:  hdds$trestbps
W = 0.96611, p-value = 1.673e-06
```

Presenta distribución normal.

Thalach

```
> qqnorm(hdds$thalach)
> qqline(hdds$thalach)
```

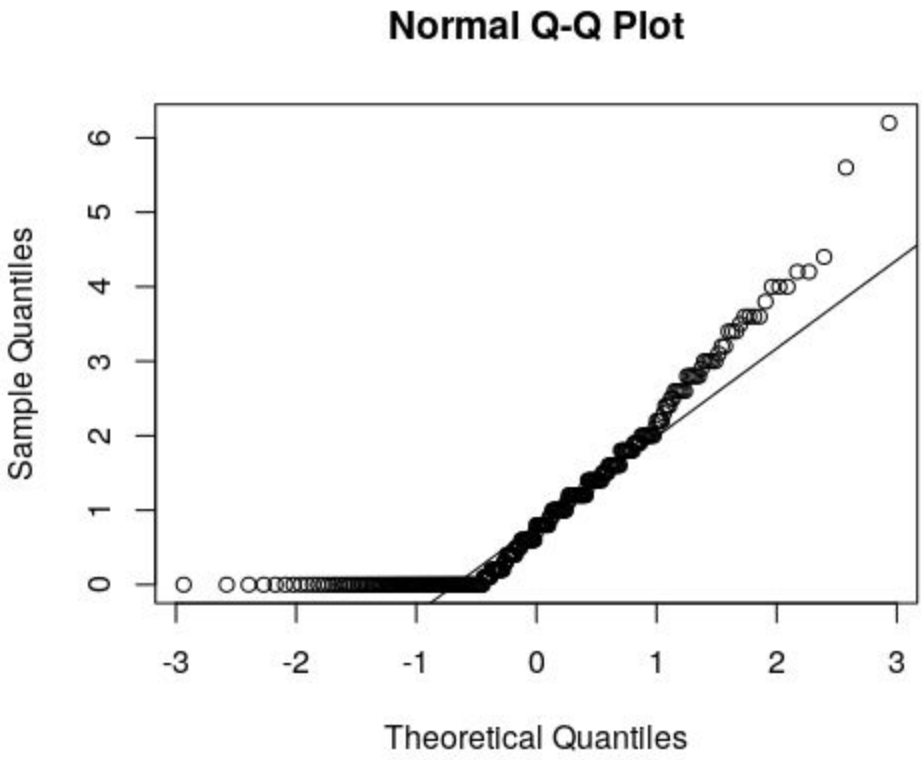



```
> shapiro.test(hdds$thalach)
Shapiro-Wilk normality test
data:  hdds$thalach
W = 0.97663, p-value = 7.997e-05
```

También está normalmente distribuida.

Oldpeak

```
> qqnorm(hdds$oldpeak)
> qqline(hdds$oldpeak)
```




```
> shapiro.test(hdds$oldpeak)
Shapiro-Wilk normality test
data:  hdds$oldpeak
W = 0.84155, p-value < 2.2e-16
```

También está distribuida normalmente, aunque con una W peor que la del resto de variables.

Por último y para rematar vamos a utilizar la prueba de Anderson-Darling. Utilizando esta prueba tendremos que si nuestro *p-valor* es superior al nivel de significación (en este caso decidimos un $\alpha = 0.05$) entonces la variable en cuestión sigue una distribución normal.

```
> install.packages("nortest")
> library(nortest)
> alpha = 0.05
> col.names = colnames(hdds)
```

```
> for (i in 1:ncol(hdds)) {
      if (i == 1) cat("Variables que siguen una distribución normal:\n")
      if (is.integer(hdds[,i]) | is.numeric(hdds[,i])) {
            p_val = ad.test(hdds[,i])$p.value
            if (p_val < alpha) {
                  cat(col.names[i])
                  message(" - p-valor: ", p_val, " ")
            }
      }
}
Variables que siguen una distribución normal:
age - p-valor: 0.000852717535620041
trestbps - p-valor: 1.36662671637797e-06
chol - p-valor: 0.0118118476774193
thalach - p-valor: 8.55725872900946e-06
oldpeak - p-valor: 3.7e-24
```

Otra vez obtenemos el mismo resultado, así que estamos seguros de que todas las variables cuantitativas siguen una distribución gaussiana.

Homogeneidad

Vamos a comprobar la homogeneidad de varianzas entre las variables cuantitativas y los grupos de enfermos / no enfermos mediante el uso del test de *Fligner-Killeen.*

Age

```
> fligner.test(hdds$age ~ hdds$target, data = hdds)
Fligner-Killeen test of homogeneity of variances
data:  hdds$age by hdds$target
Fligner-Killeen:med chi-squared = 6.7669, df = 1, p-value = 0.009286
```

Nuestro p-valor es menor que nuestro α = 0.05 por lo que tenemos que descartar nuestra hipótesis de homogeneidad entre las varianzas de ambos grupos.

Trestbps

```
> fligner.test(hdds$trestbps ~ hdds$target, data = hdds)
Fligner-Killeen test of homogeneity of variances
data:  hdds$trestbps by hdds$target
Fligner-Killeen:med chi-squared = 1.555, df = 1, p-value = 0.2124
```

En este caso el p-valor es mayor que α por lo que podemos asegurar que las varianzas de ambas muestras son homogéneas.

Chol

```
> fligner.test(hdds$chol ~ hdds$target, data = hdds)
Fligner-Killeen test of homogeneity of variances
data:  hdds$chol by hdds$target
Fligner-Killeen:med chi-squared = 0.99461, df = 1, p-value = 0.3186
```

Igual que en el caso anterior tenemos varianzas homogéneas entre grupos.

Thalach

```
> fligner.test(hdds$thalach ~ hdds$target, data = hdds)
Fligner-Killeen test of homogeneity of variances
data:  hdds$thalach by hdds$target
Fligner-Killeen:med chi-squared = 5.026, df = 1, p-value = 0.02497
```

Tenemos un p-valor menor que α. Descartamos la hipótesis.

Oldppeak

```
> fligner.test(hdds$oldpeak ~ hdds$target, data = hdds)
Fligner-Killeen test of homogeneity of variances
data:  hdds$oldpeak by hdds$target
Fligner-Killeen:med chi-squared = 34.238, df = 1, p-value = 4.877e-09
```

En este caso también descartamos la hipótesis.

Todas las cuantitativas

Vamos a calcular la homogeneidad de todas las cuantitativas combinadas, a ver qué nos sale.

```
> fligner.test(hdds$age + hdds$trestbps + hdds$chol + hdds$thalach +
hdds$oldpeak~ hdds$target, data = hdds)
Fligner-Killeen test of homogeneity of variances
data:  hdds$age + hdds$trestbps + hdds$chol + hdds$thalach + hdds$oldpeak by
hdds$target
Fligner-Killeen:med chi-squared = 0.53239, df = 1, p-value = 0.4656
```

En este caso tenemos que la combinación lineal de todas las variables cuantitativas sí presenta una homogeneidad de varianzas en cuanto a los grupos de enfermos y sanos.

Pruebas estadísticas

Análisis de correlación

Vamos a hacer un análisis de correlación para ver qué variables son las que más influyen en la presencia o ausencia de enfermedad cardíaca. Primero vamos a comprobar las correlaciones existentes entre los datos de tipo numérico.

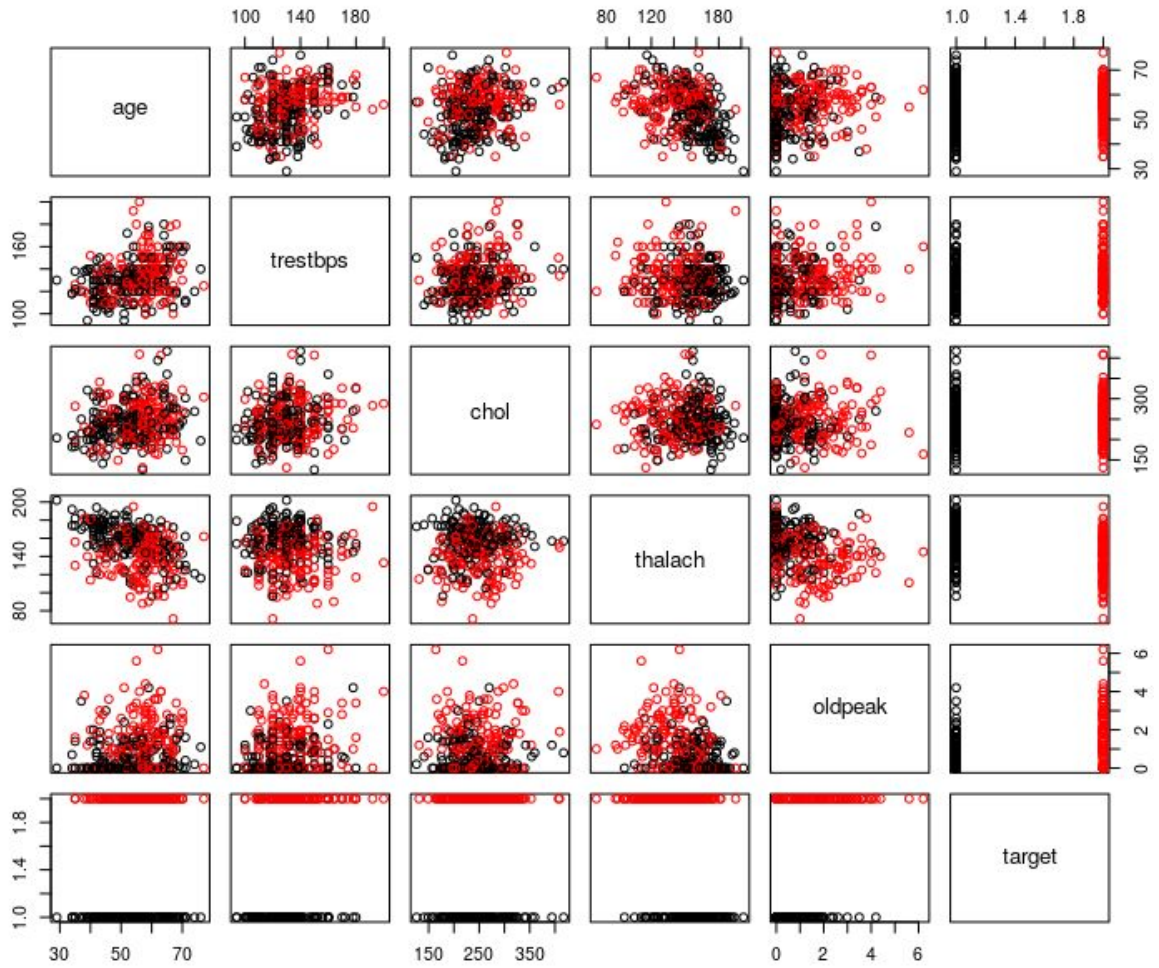
```
> install.packages("corrplot")
> library(corrplot)
> hddsNum <- subset(hdds, select=c(age, trestbps, chol, thalach, oldpeak,
target),)
> hddsNum[,c("target")] <- as.numeric(hdds[,c("target")])
> corrplot.mixed(cor(hddsNum), lower = "number", upper = "square")
```

Todos los atributos correlacionan con algo con la edad, inversamente en el caso de *thalach*, que es la máxima frecuencia cardiaca obtenida. Además tenemos una correlación inversa entre *thalach* y *oldpeak*, esta última definida como el descenso de la ST inducida por el ejercicio en relación al reposo.

Los atributos que más correlacionan con la enfermedad son, por orden: *oldpeak, thalach, age, trestbps y chol*.

Vamos a graficar todas las combinaciones de variables superponiendo los colores de *target* a ver si distinguimos alguna separación notable entre los grupos.

```
> plot(hddsNum, col=hdds$target)
```



Salvo para el caso de *target*, como era de esperar, no se visualiza gran separación entre grupos.

En este primer análisis hemos omitido todas las variables categóricas. Para este tipo lo más usual es hacer un test de Chi-cuadrado, pero de esta manera relacionaremos también las variables categóricas con las numéricas a ver qué encontramos.

Lo que vamos a hacer, de manera poco ortodoxa, es convertir todos los factores a variables numéricas bajo la premisa de que no deberíamos perder información en la transformación.

```
> hddsB <- hdds
> for (i in c("sex", "cp", "fbs", "restecg", "exang", "slope", "ca", "thal",
"target")) {
      hddsB[,i] <- as.numeric(hdds[,i])
  }
> corrplot.mixed(cor(hddsB), lower = "number", upper = "square")
```

Según estos resultados los atributos más correlacionados son *oldpeak* y *slope* con un coeficiente de 0.57.

Por defecto el *correlation plot* calcula mediante el método de *Pearson* así que lo vamos a complementar con el de *Spearman* focalizándonos en la variable *target*, que es la que nos interesa observar.

```
> corRes <- cor(hddsB, use = "complete.obs", method = "spearman")
```

```
> corRes["target",]
age          sex          cp     trestbps     chol         fbs    restecg
thalach      exang
0.24754258  0.27642092  0.46945674  0.12874183  0.12670392  0.03170974
0.17665022 -0.42403220  0.42953493
oldpeak      slope        ca     thal   target
0.42188878  0.37602538  0.48556253  0.53312479  1.00000000
```

Podemos observar cierta correlación de variables, siendo las que *menos* influyen en la enfermedad el colesterol y el nivel de azúcar en sangre. En cualquier caso ninguna presenta una correlación abrumadora, siendo todas menores que 0.6.

Contraste de Hipótesis

Vamos a realizar un contraste de hipótesis enunciado de la siguiente manera: *los niveles de colesterol en personas con enfermedad coronaria son diferentes a los individuos sanos.*

Nuestra hipótesis queda enunciada de la siguiente manera:

H0: μE = μS
H1: μE <> μS (bilateral)

Ya hemos comprobado que la variable *chol* se ajusta a una distribución normal lo que implica que se trata de un test paramétrico. Como no conocemos la varianza poblacional tendremos que usar distribuciones de t-Student. Además, al tratarse de un contraste de dos muestras debemos comprobar si se puede asumir que las varianzas son iguales o no (homocedasticidad). Para ello esta vez utilizaremos el test *F de Snedecor.*

```
> hddsEnfermo <- hdds[hdds$target==1,]$chol
> hddsSano <- hdds[hdds$target==0,]$chol
> var.test(hddsEnfermo, hddsSano, alternative = "two.sided", conf.level = 0.95)
F test to compare two variances
data:  hddsEnfermo and hddsSano
F = 1.0897, num df = 138, denom df = 161, p-value = 0.5979
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.790682 1.507905
sample estimates:
ratio of variances
      1.089699
```

Como p-valor > α no podemos rechazar la hipótesis nula. Por lo tanto suponemos homogeneidad de varianzas.

Ahora vamos a realizar el test en sí, con una significación del 95%.

```
> t.test(chol~target, alternative='two.sided', conf.level=.95, var.equal=TRUE,
data=hdds)
Two Sample t-test
data:  chol by target
t = -1.9235, df = 299, p-value = 0.05536
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -21.7658719   0.2483304
sample estimates:
mean in group 0 mean in group 1
      240.7160    251.4748
```

A la vista de los resultados tenemos que el p-valor = 0.05536 es ligeramente mayor que nuestro α = 0.05 por lo que tenemos que concluir que no se puede rechazar H0. Esto implica que los niveles de colesterol entre individuos sanos y enfermos **no** son distintos desde un punto de vista de significancia estadística.

Regresión Logística

Vamos a intentar realizar predicciones sobre la presencia de enfermedad cardíaca en función del resto de variables contenidas en nuestro dataset. Así, se calculará un modelo de regresión logística utilizando regresores tanto cuantitativos como cualitativos con el que poder realizar las predicciones de enfermedad.

Para obtener un modelo de regresión considerablemente eficiente, lo que haremos será obtener varios modelos utilizando las variables que estén más correladas con respecto a la presencia de enfermedad según las correlaciones ya estudiadas.

También haremos regresión sobre el Test de Riesgo de Framingham [5] aunque como no tenemos todas las variables que se necesitan vamos a simplificarlo y a incluir únicamente la edad *(age)*, el nivel de colesterol *(chol)* y la presión sanguínea *(trestbps).* Es curioso que el test de Framingham tiene en cuenta el colesterol antes que otras variables que en nuestro estudio correlacionan más con la enfermedad cardíaca, pero no voy a venir yo a enmendarle la plana a la comunidad médica.

Así, de entre todos los modelos que tengamos, escogeremos el mejor utilizando como criterio aquel que presente un menor factor AIC (Akaike's Information Criteria).

Regresión logística múltiple cuantitativa basada en Framingham

```
> hddsFram <- hddsNum[,c("age", "chol", "trestbps", "target")]
> corRes <- cor(hddsFram, use = "complete.obs")
> corRes
              age    chol  trestbps     target
age    1.0000000 0.1948723 0.2890095 0.2320268
chol   0.1948723 1.0000000 0.1607868 0.1105587
trestbps 0.2890095 0.1607868 1.0000000 0.1510304
target   0.2320268 0.1105587 0.1510304 1.0000000
> corrplot.mixed(corRes, lower = "ellipse", upper = "square")
```


```
> lrFram <- glm(hdds$target ~ age+chol+trestbps, hddsFram, family=binomial)
> summary(lrFram)
Call:
glm(formula = hdds$target ~ age + chol + trestbps, family = binomial,
      data = hddsFram)

Deviance Residuals:
      Min    1Q    Median        3Q     Max
-1.6380  -1.0657  -0.7808   1.1450   1.7576
```

```
Coefficients:
             Estimate Std. Error z value  Pr(>|z|)
(Intercept) -4.657283   1.150294  -4.049 0.0000515 ***
age          0.046281   0.014422   3.209   0.00133 **
chol         0.002609   0.002527   1.033   0.30183
trestbps     0.010137   0.007166   1.415   0.15716
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

      Null deviance: 415.52  on 300  degrees of freedom
Residual deviance: 395.44  on 297  degrees of freedom
AIC: 403.44

Number of Fisher Scoring iterations: 4
```

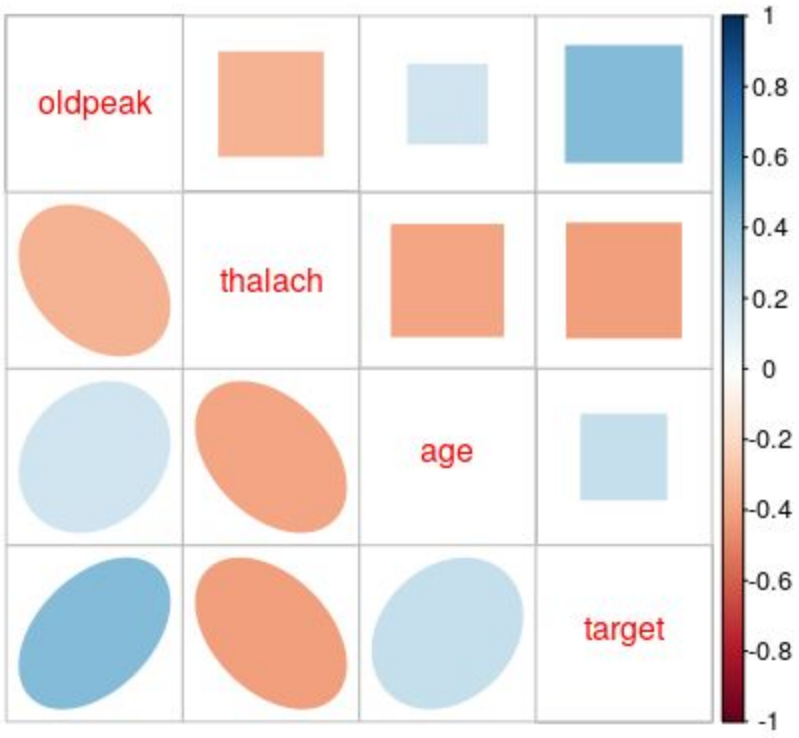
Obtenemos unos códigos de significación de dos asteriscos para la edad y de nada para *chol y trestbps*, lo que era de esperar a la vista de las correlaciones. Conseguimos un AIC de 403.44, que nos servirá para comparar con otros modelos.

Regresión logística múltiple cuantitativa basada en correlaciones

Para este modelo vamos a utilizar las tres variables cuantitativas que más correlacionan con la enfermedad que hemos encontrado que son *oldpeak, thalach y age*.

```
> hddsCor <- hddsNum[,c("oldpeak", "thalach", "age", "target")]
> corRes <- cor(hddsCor, use = "complete.obs")
> corRes
            oldpeak     thalach      age    target
oldpeak   1.0000000 -0.3448108  0.1995496  0.4315440
thalach -0.3448108  1.0000000 -0.3979785 -0.4170412
age    0.1995496 -0.3979785  1.0000000  0.2320268
target   0.4315440 -0.4170412  0.2320268  1.0000000
```

```
> corrplot.mixed(corRes, lower = "ellipse", upper = "square")
```




```
> lrCor <- glm(hdds$target ~ oldpeak+thalach+age, hddsCor, family=binomial)
> summary(lrCor)
Call:
glm(formula = hdds$target ~ oldpeak + thalach + age, family = binomial,
      data = hddsCor)

Deviance Residuals:
      Min    1Q    Median        3Q     Max
-2.2360  -0.8591  -0.5005   0.8331   2.1925

Coefficients:
            Estimate Std. Error z value    Pr(>|z|)
(Intercept)  2.782870   1.599505   1.740    0.0819 .
oldpeak      0.739119   0.141792   5.213 0.000000186 ***
thalach      -0.030711   0.007047  -4.358 0.000013116 ***
age          0.016612   0.016323   1.018    0.3088
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
      Null deviance: 415.52  on 300  degrees of freedom
Residual deviance: 324.00  on 297  degrees of freedom
AIC: 332

Number of Fisher Scoring iterations: 4
```

En este caso tenemos tres asteriscos de significación para *oldpeak y thalach* y ninguno para *age.* Con un AIC de 332 podemos asegurar que este modelo representa mejor que el anterior la enfermedad cardíaca.

Regresión logística múltiple cualitativa basada en correlaciones

En este caso vamos a elegir las tres variables que más correlacionan con la enfermedad, que hemos encontrado que son las de tipo cualitativo *thal, ca y cp.*

```
> hddsCor2 <- hdds[,c("thal", "ca", "cp", "target")]
> lrCor2 <- glm(hdds$target ~ thal+ca+cp, hddsCor2, family=binomial)
> summary(lrCor2)
Call:
glm(formula = hdds$target ~ thal + ca + cp, family = binomial,
      data = hddsCor2)

Deviance Residuals:
      Min    1Q   Median        3Q     Max
-2.7700  -0.7218  -0.2812   0.6778   2.5497

Coefficients:
                  Estimate Std. Error z value    Pr(>|z|)
(Intercept)        -2.5475     0.5538  -4.600 0.0000042180 ***
thal[T.chronic]    1.4828      0.6629   2.237     0.025297 *
thal[T.reversible]   2.2484    0.3595   6.254 0.0000000004 ***
ca[T.1.0]          1.9988      0.4192   4.768 0.0000018583 ***
ca[T.2.0]          2.3502      0.5779   4.067 0.0000476093 ***
ca[T.3.0]          2.5626      0.7766   3.300     0.000968 ***
cp[T.atypical]     -0.4124     0.6520  -0.633     0.527034
cp[T.non-anginal]   -0.6634    0.6157  -1.077     0.281330
cp[T.asymptomatic]   1.5511    0.5487   2.827     0.004697 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

      Null deviance: 415.52  on 300  degrees of freedom
Residual deviance: 231.28  on 292  degrees of freedom
AIC: 249.28
```

```
Number of Fisher Scoring iterations: 5
```

Podemos ver que la mayor significancia a la hora de predecir la enfermedad cardíaca con estas variables es la presencia de talasemia reversible, el tener 1 o más vasos sanguíneos principales coloreados por fluoroscopia y el dolor de pecho asintomático.

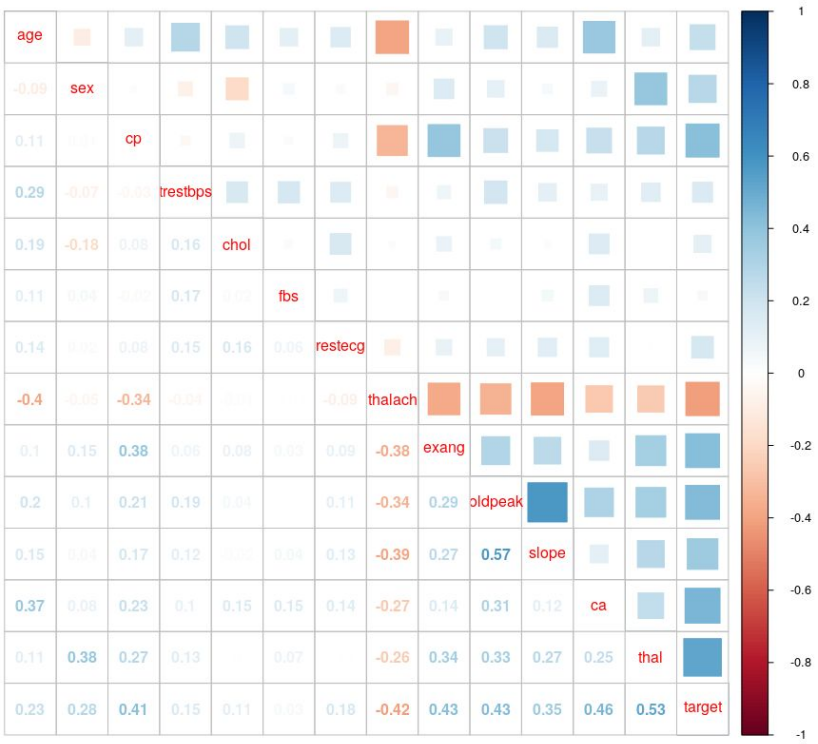
Además conseguimos un factor AIC de 249.28, que es el mejor de los obtenidos hasta ahora.

Podríamos seguir añadiendo variables y mejorando el modelo, pero para esta práctica nos conformaremos con tres modelos de regresión.

Resultados

Además de todo lo que ya se ha expuesto vamos a resumir en este apartado los resultados de nuestros análisis.

Análisis de correlación



Contraste de hipótesis

Hipótesis:

H0: μE = μS
H1: μE <> μS (bilateral)

Resultados:

```
F test to compare two variances
data:  hddsEnfermo and hddsSano
F = 1.0897, num df = 138, denom df = 161, p-value = 0.5979
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.790682 1.507905
sample estimates:
ratio of variances
```

Regresión logística

Variables age, chol y trestbps

```
glm(formula = hdds$target ~ age + chol + trestbps, family = binomial,
      data = hddsFram)

Deviance Residuals:
      Min    1Q   Median        3Q     Max
-1.6380  -1.0657  -0.7808   1.1450   1.7576

Coefficients:
             Estimate Std. Error z value  Pr(>|z|)
(Intercept) -4.657283   1.150294  -4.049 0.0000515 ***
age          0.046281   0.014422   3.209   0.00133 **
chol         0.002609   0.002527   1.033   0.30183
trestbps     0.010137   0.007166   1.415   0.15716
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
      Null deviance: 415.52  on 300  degrees of freedom
Residual deviance: 395.44  on 297  degrees of freedom
AIC: 403.44
Number of Fisher Scoring iterations: 4
```

Variables oldpeak, thalach y age

```
glm(formula = hdds$target ~ oldpeak + thalach + age, family = binomial,
      data = hddsCor)

Deviance Residuals:
      Min    1Q   Median        3Q     Max
-2.2360  -0.8591  -0.5005   0.8331   2.1925

Coefficients:
             Estimate Std. Error z value    Pr(>|z|)
(Intercept)  2.782870   1.599505   1.740    0.0819 .
oldpeak      0.739119   0.141792   5.213 0.000000186 ***
thalach     -0.030711   0.007047  -4.358 0.000013116 ***
age          0.016612   0.016323   1.018    0.3088
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)
      Null deviance: 415.52  on 300  degrees of freedom
Residual deviance: 324.00  on 297  degrees of freedom
AIC: 332
Number of Fisher Scoring iterations: 4
```

Variables thal, ca y cp

```
glm(formula = hdds$target ~ thal + ca + cp, family = binomial,
      data = hddsCor2)

Deviance Residuals:
      Min    1Q   Median        3Q    Max
-2.7700  -0.7218  -0.2812   0.6778   2.5497

Coefficients:
                  Estimate Std. Error z value    Pr(>|z|)
(Intercept)        -2.5475     0.5538  -4.600 0.0000042180 ***
thal[T.chronic]    1.4828      0.6629   2.237    0.025297 *
thal[T.reversible]   2.2484    0.3595   6.254 0.0000000004 ***
ca[T.1.0]          1.9988      0.4192   4.768 0.0000018583 ***
ca[T.2.0]          2.3502      0.5779   4.067 0.0000476093 ***
ca[T.3.0]          2.5626      0.7766   3.300    0.000968 ***
cp[T.atypical]     -0.4124     0.6520  -0.633    0.527034
cp[T.non-anginal]   -0.6634    0.6157  -1.077    0.281330
cp[T.asymptomatic]   1.5511    0.5487   2.827    0.004697 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
      Null deviance: 415.52  on 300  degrees of freedom
Residual deviance: 231.28  on 292  degrees of freedom
AIC: 249.28
Number of Fisher Scoring iterations: 5
```

Resumen

Regresión logística sobre variables	**AIC**
age, chol y trestbps	403.44
oldpeak, thalach y age	332
thal, ca y cp	249.28

Conclusiones

Al igual que en el apartado anterior vamos a resumir a continuación las conclusiones de nuestros análisis, aunque ya han sido expuestas con anterioridad.

Análisis de correlación

Todos los atributos correlacionan con algo con la edad, inversamente en el caso de *thalach*, que es la máxima frecuencia cardiaca obtenida. Además tenemos una correlación inversa entre *thalach* y *oldpeak*, esta última definida como el descenso de la ST inducida por el ejercicio en relación al reposo.

Los atributos más correlacionados entre si son *oldpeak* y *slope* con un coeficiente de 0.57.

Utilizando Pearson los que más correlacionan con la enfermedad son, por orden: *oldpeak, thalach, age, trestbps y chol*.

En el test de Spearman obtenemos cierta correlación de variables, siendo las que *menos* influyen en la enfermedad el colesterol y el nivel de azúcar en sangre. En cualquier caso ninguna presenta una correlación abrumadora, siendo todas menores que 0.6.

Contraste de hipótesis

Formulamos la hipótesis de que los niveles de colesterol en personas con enfermedad coronaria son diferentes a los individuos sanos.

Nuestros resultados indican que los niveles de colesterol entre individuos sanos y enfermos no son distintos desde un punto de vista de significancia estadística.

Regresión logística

Calculamos tres modelos distintos de regresión logística siendo el más útil de cara a predecir la presencia o ausencia de enfermedad coronaria el que incluye las variables *thal, ca y cp.*

CÓDIGO

El código ya ha sido presentado a lo largo de todo este documento. En cualquier caso se han compilado las ~160 líneas de código en un archivo adjunto a este documento.

Bibliografía

[1] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[2] R. Detrano. International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease
http://rexa.info/paper/b884ce2f4aff7ed95ce7bfa7adabaef46b88c60c

[3] Christopher J. O'Donnell y Roberto Elosua. Factores de riesgo cardiovascular. Perspectivas derivadas del Framingham Heart Study
http://www.revespcardiol.org/es/factores-riesgo-cardiovascular-perspectivas-derivadas/articulo/13116658/?esMedico=1

[4] Wilson PWF, D'Agostino RB, Levy D, Belanger AM, Silbers-hatz H, Kannel WB. Prediction of coronary heart disease using risk factor categories. Circulation. 1998;97:1837-47.

[5] https://en.wikipedia.org/wiki/Framingham_Risk_Score

[6] Eichler K, Puhan MA, Steurer J, Bachmann LM. Prediction of first coronary events with the Framingham score: A systematic review. Am Heart J. 2007;153:722-31.
https://www.semanticscholar.org/paper/Prediction-of-first-coronary-events-with-the-score%3A-Eichler-Puhan/70521d45bb6d3a6f16df618fc06257701187818a

[7] http://www.unige.ch/ses/sococ/cl/r/tasks/outliers.e.html

[8]
https://www.doctoralia.es/preguntas-respuestas/con-una-tension-de-200-100-que-es-lo-correcto-a-hacer-ponerse-a-caminar-o-mejor-estar-sentada-para