PEC2

Pablo Monforte Izquierdo

2022-12-21

Contents

1. Descripción del dataset

¿Por qué es importante y qué pregunta/problema pretende responder?

El conjunto de datos objeto de análisis se ha obtenido a partir de este enlace en Kaggle y está constituido por 14 características (columnas) que presentan 303 pacientes (filas o registros). Entre los campos de este conjunto de datos, encontramos los siguientes:

- Age : Edad del paciente
- Sex : Sexo del paciente
- cp : Tipo de dolor torácico tipo de dolor torácico
  - Value 1: angina típica
  - Value 2: angina atípica
  - Value 3: dolor no anginoso
  - Value 4: asintomático
- trtbps : presión arterial en reposo (en mm Hg)
- chol : colesterol en mg/dl obtenido a través del sensor de IMC
- fbs : (glucemia en ayunas > 120 mg/dl) (1 = verdadero; 0 = falso)

- rest_ecg : resultados electrocardiográficos en reposo
  - Value 0: normal
  - Value 1: con anomalía de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST de > 0,05 mV)
  - Value 2: hipertrofia ventricular izquierda probable o definida según los criterios de Estes
- thalach : frecuencia cardiaca máxima alcanzada
- exng: angina inducida por ejercicio (1 = sí; 0 = no)
- oldpeak: Pico anterior
- slp: Pendiente
- ca: número de vasos principales (0-3)
- thall: Ratio Thal
- output: 0= menor probabilidad de infarto 1= mayor probabilidad de infarto

Con este conjunto se plantea la problemática de determinar qué variables influyen más a la hora de tener un infarto. También se construirán modelos de regresión logistica para determinar si una persona sufrirá un infarto o no. Además también se harán contrastes de hipotesis para detectar propiedades interesantes en las muestras que puedan ser inferidas con respecto a la población.

Estos analisis son muy importantes para la salud de las personas ya que puede ayudar a detectar que personas tienen un riesgo alto de sufrir un infarto.

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

Puede ser el resultado de adicionar diferentes datasets o una subselección útil de los datos originales, en base al objetivo que se quiera conseguir.

Para resolver el problema solamente se utilizará el siguinte dataset. Después de realizar el analisis correspondiente es posible que algunas variables de descarten a la hora de realizar los análisis.

```
heart = read.csv('./heart.csv')
dim(heart)
```

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

```
head(heart)
```

```
##   age sex cp trtbps chol fbs restecg thalachh exng oldpeak slp caa thall output
## 1  63   1  3    145  233   1       0      150    0     2.3   0   0     1      1
## 2  37   1  2    130  250   0       1      187    0     3.5   0   0     2      1
## 3  41   0  1    130  204   0       0      172    0     1.4   2   0     2      1
## 4  56   1  1    120  236   0       1      178    0     0.8   2   0     2      1
## 5  57   0  0    120  354   0       1      163    1     0.6   2   0     2      1
## 6  57   1  0    140  192   0       1      148    0     0.4   1   0     1      1
```

3. Limpieza de los atos

Lo primero que haremos será ver el tipo de dato de cada columna para ver si necesitamos realizar modificaciones.

```
str(heart)
```

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

Vemos que todos los valores son tipo int o num pero hay algunos que nos interesa que sean tipo factor ya que son algunas variables categóricas con un número finito de valores o niveles.

```
heart$sex <- as.factor(heart$sex)
heart$cp <- as.factor(heart$cp)
heart$restecg <- as.factor(heart$restecg)
heart$exng <- as.factor(heart$exng)
heart$thall <- as.factor(heart$thall)
heart$output <- as.factor(heart$output)
```

Vemos que ya tenemos los datos con los formatos que nos interesan.

```
str(heart)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 2 ...
##  $ cp      : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ 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 : Factor w/ 3 levels "0","1","2": 1 2 1 2 2 2 1 2 2 2 ...
##  $ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng    : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ 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   : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
##  $ output  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

3.1. ¿Los datos contienen ceros o elementos vacíos?

Gestiona cada uno de estos casos.

A continuación vamos a comprobar si los datos tienen valores nulos.

```
sapply(heart, function(x) sum(is.na(x)))
```

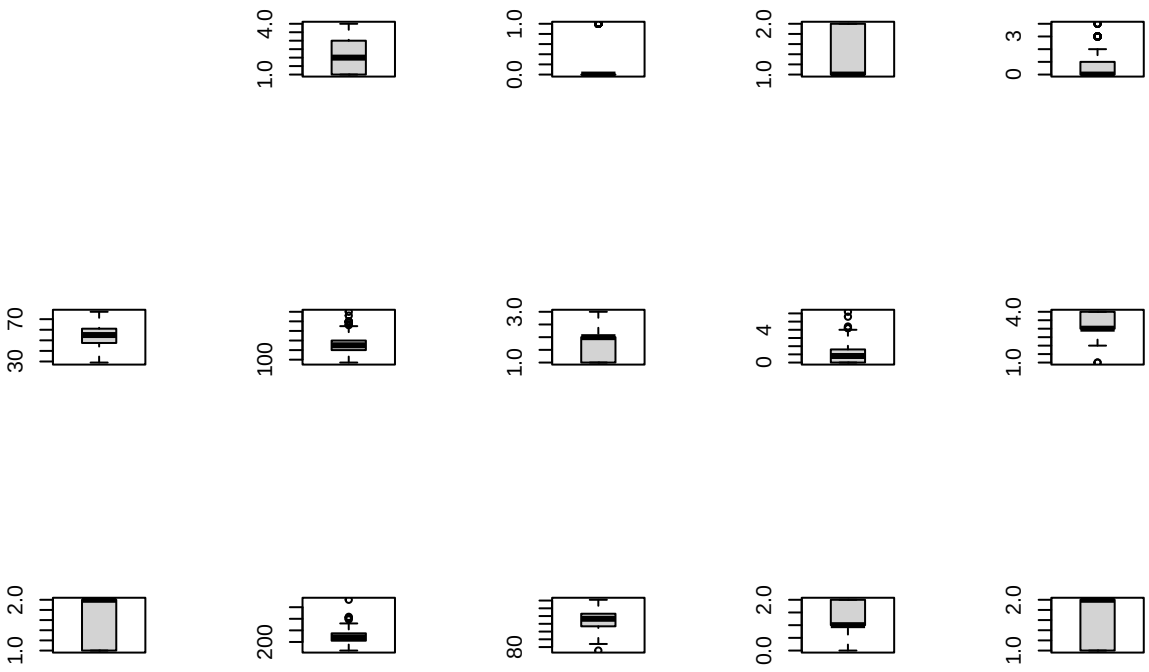
```
##      age      sex       cp   trtbps     chol      fbs  restecg thalachh
##        0        0        0        0        0        0        0        0
##     exng  oldpeak      slp      caa    thall   output
##        0        0        0        0        0        0
```

Vemos que no existe ningún valor nulo por lo que no tendremos que gestionarlos.

3.2. Identifica y gestiona los valores extremos

A continuación vamos a ver si existen valores extremos en las variables.

```
layout(matrix(c(0:14), nrow=3, byrow=FALSE))
for (i in 1:14) boxplot(heart[i])
```



Vemos que tenemos outliers en trtbps, chol, fbs, thalachh, oldpeak, caa, thall. A continuación vamos a ver que outliers elimnaremos y cuales no.

Los outliers de trtbps no parecen que sean datos erroneos si no mediciones que son poco normales por lo que no los eliminaremos.

Con los outliers de chol ocurre lo mismo que con los de trtbps.

En fbs no hay outilers solamente pocos valores con 1.

En thalachh pasa lo mismo que el chol y trtbps.

En oldpeak sucede lo mismo.

En caa si que tenemos outlier erroneos ya que en algunos registros tenemos el valor 4 y esto es imposible ya que solamente podemos tener valores entre 0 y 3 por lo que eliminaremos estos registros.

```
heart <- heart[heart$caa <= 3, ]
```

Para thall no parecen que sean datos erroneos si no mediciones que son poco normales por lo que no los eliminaremos.

Ya tenemos los datos listos para anlizar por lo que exportaremos el archivo limpio.

```
write.csv(heart, "heart_clean.csv")
```

4. Análisis de los datos

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

(p.ej., si se van a comparar grupos de datos, ¿cuáles son estos grupos y qué tipo de análisis se van a aplicar?)

Vamos a selecionar los grupos que vamos a estudiar.

```
hombres <- heart[heart$sex == 1, ]
mujeres <- heart[heart$sex == 0, ]

pacientes_riesgo <- heart[heart$output == 1, ]
paceintes_no_riesgo <- heart[heart$output == 0, ]
```

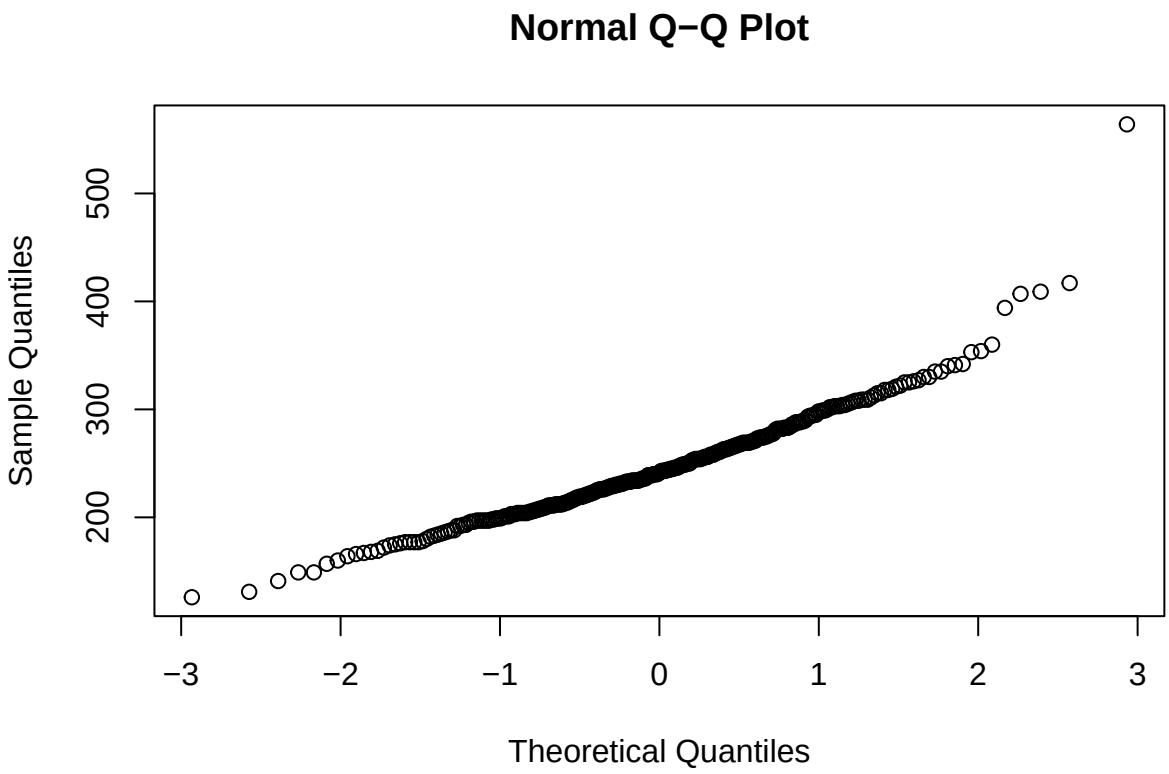
4.2. Comprobación de la normalidad y homogeneidad de la varianza

Comprobaremos la normalida y homogeneidad de la variaznza de la variables chol ya que en en los siguientes apartados haremos contrastes de hipotesis en torno a esta variable.

```
shapiro.test(heart$chol)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  heart$chol
## W = 0.94696, p-value = 6.896e-09
```

```
qqnorm(heart$chol)
```

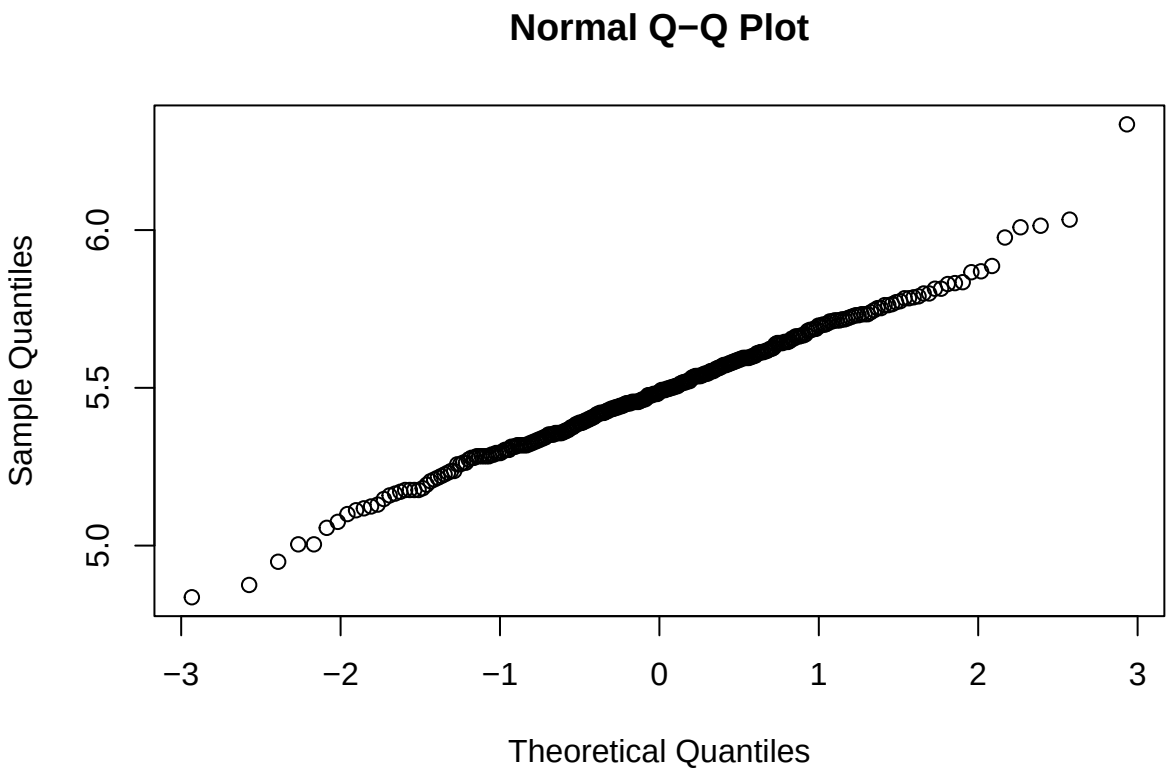


Vemos que la variable chol no sigue una distribución normal. Por lo que tendremos que aplicar boxCox para que la varibale siga una distribución normal y después volveremos a comprobar si ya sigue una distribución normal.

```
heart$chol_norm <- log(heart$chol)
shapiro.test(heart$chol_norm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  heart$chol_norm
## W = 0.99195, p-value = 0.1053
```

```
qqnorm(heart$chol_norm)
```



Ahora la variable que hemos guardado en heart$chol_norm ya sigue una distribución normal.

Tenemos que volver a seleccionar nuestros grupos de interés con la nueva variable.

```
hombres <- heart[heart$sex == 1, ]
mujeres <- heart[heart$sex == 0, ]

pacientes_riesgo <- heart[heart$output == 1, ]
paceintes_no_riesgo <- heart[heart$output == 0, ]
```

Ahora vamos a ver si los hombres y mujeres siguen tienen homogeneidad de la varianza respecto al colesterol.

```
fligner.test(x = list(hombres$chol_norm,mujeres$chol_norm))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(hombres$chol_norm, mujeres$chol_norm)
## Fligner-Killeen:med chi-squared = 5.5336, df = 1, p-value = 0.01865
```

Vemos que tienen los hombres y mujeres tienen varianzas homogeneas respecto a la variable colesterol normalizada.

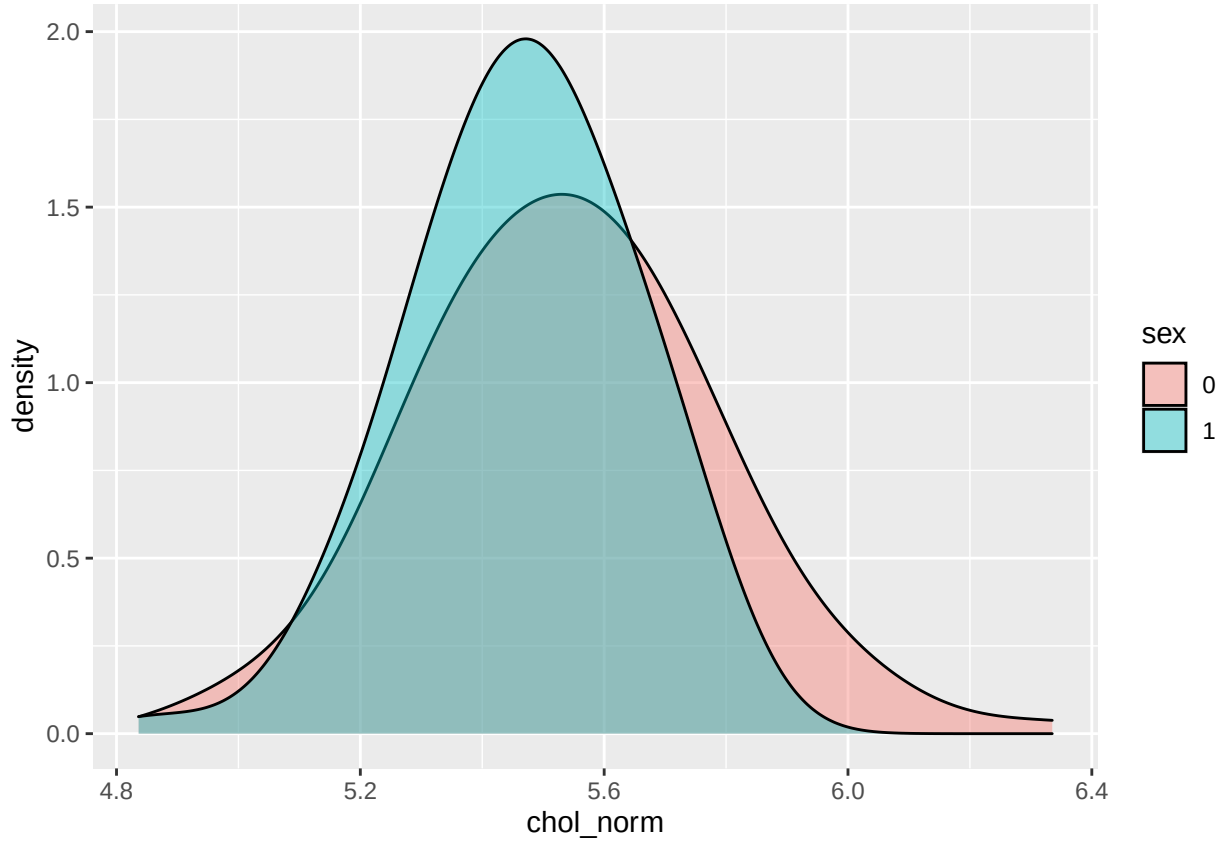
4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos

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.

¿Ser mujer influye en los niveles de colesterol?

Lo primero que vamos a hacer va a ser visualizar los datos para entenderlos mejor.

```
library(ggplot2)
ggplot(data=heart, aes(x=chol_norm,  fill=sex)) +
    geom_density(adjust=1.5, alpha=.4)
```



Hipotesis:

La hipotesis nula es que las medias de colesterol de los hombres y mujeres sean iguales:

H0:

$$\mu 1 = \mu 2$$

La hiposesis altenativa es que las medias de colesterol de los hombres y mujeres no sean iguales:

H1:

$$\mu 1 \neq \mu 2$$

Como va variable colesterol sigue una distribución normal pero desconocemos su varianza haremos un contraste t de dos colas, pero antes tenemos que saber si las varianzas son iguales o diferetes.

```
var.test(hombres$chol_norm, mujeres$chol_norm)
```

```
##
##  F test to compare two variances
```

```
##
## data:  hombres$chol_norm and mujeres$chol_norm
## F = 0.60513, num df = 201, denom df = 95, p-value = 0.003259
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.423174 0.846690
## sample estimates:
## ratio of variances
##           0.605134
```

Las varianzas son desiguales, ahora que ya tenemos toda la información necesaria podemos realizar el contraste.

```
t.test(hombres$chol_norm, mujeres$chol_norm,
       alternative = "two.sided",
       var.equal = FALSE,
       conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  hombres$chol_norm and mujeres$chol_norm
## t = -2.6815, df = 151.57, p-value = 0.008141
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.12681000 -0.01921785
## sample estimates:
## mean of x mean of y
##  5.464636  5.537650
```

Vemos que el p-valor es de 0.008141 y como es menor que alfa que es 0.05 rechazamos la h0. Esto quiere decir que las medias de los nieveles de colesterol de los hombres y mujeres no sean iguales.

¿Las mujeres tienen más colesterol que los hombres?

Ahora el contraste que realizaremos será el siguiente.

Hipotesis:

La hipotesis nula es que las medias de colesterol de los hombres y mujeres sean iguales:

H0:

$$\mu 1 = \mu 2$$

La hiposesis altenativa es que la media de colesterol de los hombres sea superior a la de las mujeres:

H1:

$$\mu 1 > \mu 2$$

Para este test utilizaremos la variable que hemos creado anteriormente de chol_norm, que ya sabemos que sigue una distribución normal y las varianzas entre hombres y mujeres son iguales.

```
t.test(mujeres$chol_norm ,hombres$chol_nor,
       alternative = "greater",
       var.equal = FALSE,
       conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  mujeres$chol_norm and hombres$chol_nor
## t = 2.6815, df = 151.57, p-value = 0.00407
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.02795188        Inf
## sample estimates:
## mean of x mean of y
##  5.537650  5.464636
```

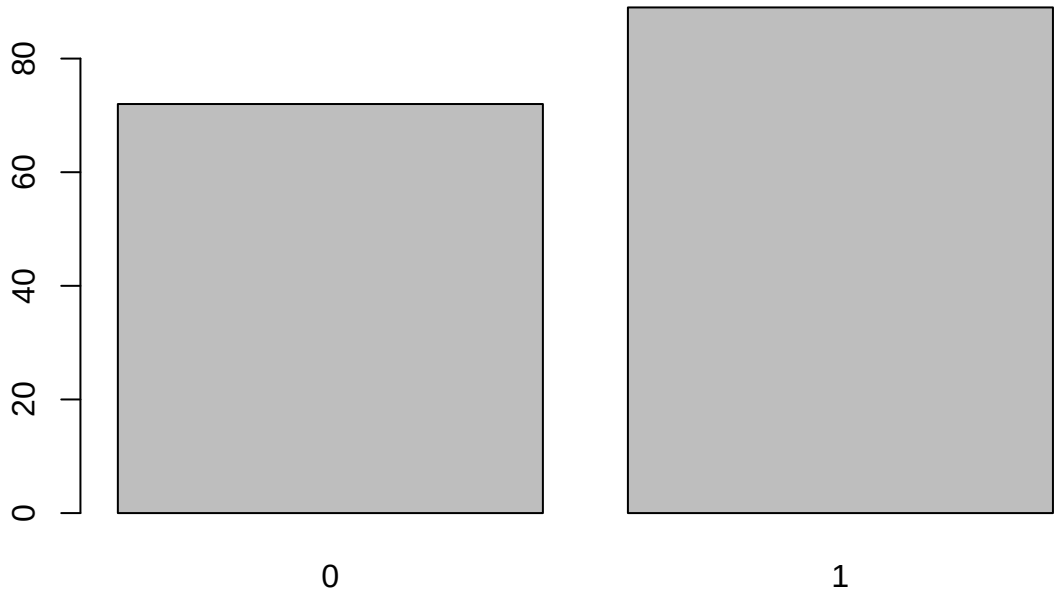
Obtenemos un p-valor inferior a 0.05 por lo que se rechaza h0. Los hombres tienen una media superior de colesterol que la de las mujeres.

¿Hay más ataques al corazon entre los hombres que entre las mujeres?

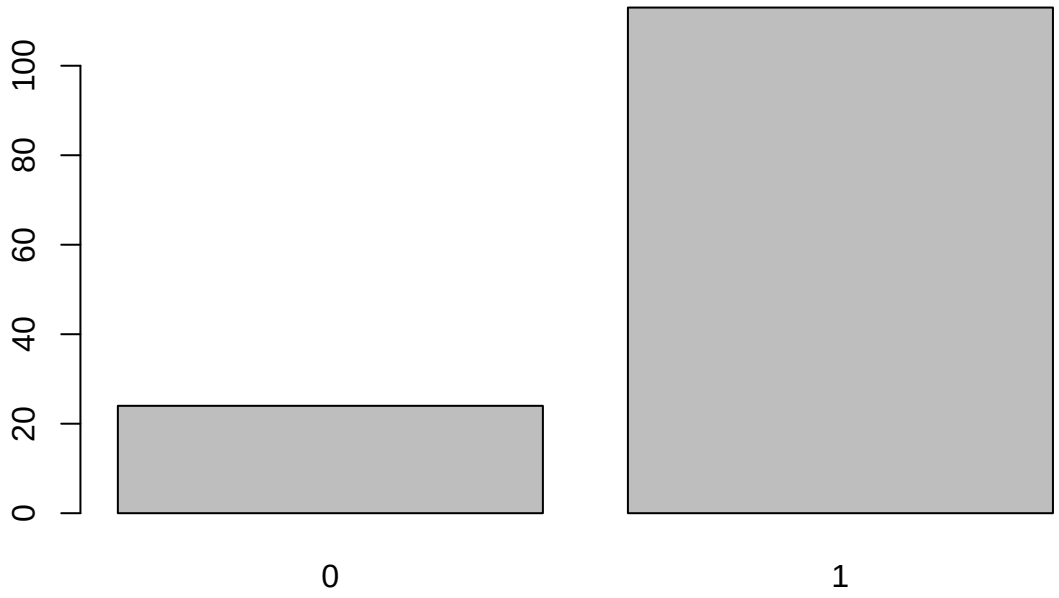
Por último la pregunta que nos haremos será si hay una proporción mayor de ataques al corazon entre los hombres que entre las mujeres.

Vamos a hacer va a ser visualizar los datos para entenderlos mejor.

```
#división entre sexos de personas con altas probablidades de sufrir un infarto
plot(factor(heart$sex[heart$output==1]))
```



```r
#división entre sexos de personas con bajas probablidades de sufrir un infarto
plot(factor(heart$sex[heart$output==0]))
```



Las hipotesis serían las siguientes:

La hipotesis nula es que la proporción de ataques al corazón es igual en hombres que en mujeres :

H0: p1 = p2

La hiposesis altenativa es que la proporción de ataques al corazón es superior en hombres que en mujeres:

H1: p1 > p2

Creamos los valores para calcular las proporciones

```r
#mujeres que pueden sufrir un infarto
minf <- dim(subset(heart, heart$sex==0 & heart$output==1))[1]
#total de mujeres
m <- dim(subset(heart, heart$sex==0 ))[1]
#hombres que pueden sugrir un infarto
hinf <- dim(subset(heart, heart$sex==1 & heart$output==1))[1]
#total de hombres.
h <- dim(subset(heart, heart$sex==1 ))[1]
```

Realizamos el test

```r
prop.test(x = c(minf, hinf), n = c(m, h),
          alternative = "greater",
          conf.level = 0.95,
          correct=FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(minf, hinf) out of c(m, h)
## X-squared = 25.081, df = 1, p-value = 2.748e-07
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.2167483 1.0000000
## sample estimates:
##    prop 1    prop 2
## 0.7500000 0.4405941
```

El p-valor es menor que 0.05 por lo que rechazamos h0 y podemos concluir que la proporción de ataques al corazón es superior en hombres que en mujeres.

Modelo de regresión

Lo siguente que haremos ser crear un modelo de regresión para predecir si una persona va a sufrir un infarto o no.

Lo primero que haremos será dividir los datos en dos muestras una para entrenar al modelo y otra para testearlo.

```r
set.seed(25)
sample <- sample(c(TRUE, FALSE), nrow(heart), replace=TRUE, prob=c(0.8,0.2))
train  <- heart[sample, ]
test   <- heart[!sample, ]
dim(train)
```

```
## [1] 244  15
```

```r
dim(test)
```

```
## [1] 54 15
```

A continuación vamos a crear un primer modelo con todas las variables que tenemos actualmente.

```r
modelo <- glm(output~., data = train, family = 'binomial')
summary(modelo)
```

```
##
## Call:
## glm(formula = output ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.9262  -0.3234   0.1348   0.4674   2.8109
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.595980  26.010686   0.369 0.712184
## age          0.025380   0.029151   0.871 0.383939
## sex1        -0.939992   0.610141  -1.541 0.123410
## cp1          0.940463   0.674602   1.394 0.163287
## cp2          2.026948   0.581500   3.486 0.000491 ***
## cp3          1.325464   0.711659   1.862 0.062533 .
## trtbps      -0.002726   0.013270  -0.205 0.837262
## chol         0.008025   0.022043   0.364 0.715799
## fbs          0.683888   0.682803   1.002 0.316542
## restecg1     0.771988   0.456537   1.691 0.090844 .
## restecg2    -1.574301   2.435177  -0.646 0.517966
## thalachh     0.012316   0.012731   0.967 0.333330
## exng1       -0.699017   0.514689  -1.358 0.174421
## oldpeak     -0.236725   0.265747  -0.891 0.373042
## slp          0.923297   0.417695   2.210 0.027073 *
## caa         -1.661126   0.329575  -5.040 4.65e-07 ***
## thall1       1.998433   1.992285   1.003 0.315819
## thall2       2.808650   1.897128   1.480 0.138747
## thall3       0.848555   1.893265   0.448 0.654011
## chol_norm   -3.015366   5.698549  -0.529 0.596704
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 335.48  on 243  degrees of freedom
## Residual deviance: 148.89  on 224  degrees of freedom
## AIC: 188.89
##
## Number of Fisher Scoring iterations: 6
```

Correlación y colinealidad

Vamos a estudiar si hay algunos valores que estén correlacionados y causen algún conflicto en el modelo.

```
car::vif(modelo)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## age        1.511101  1        1.229269
## sex        1.647134  1        1.283407
## cp         1.779117  3        1.100781
## trtbps     1.318453  1        1.148239
## chol      25.349262  1        5.034805
## fbs        1.297804  1        1.139212
## restecg    1.312020  2        1.070250
## thalachh   1.609105  1        1.268505
## exng       1.275596  1        1.129423
## oldpeak    1.565948  1        1.251378
## slp        1.614177  1        1.270503
```

```
## caa         1.517787  1         1.231985
## thall       1.940756  3         1.116851
## chol_norm 24.994790  1         4.999479
```

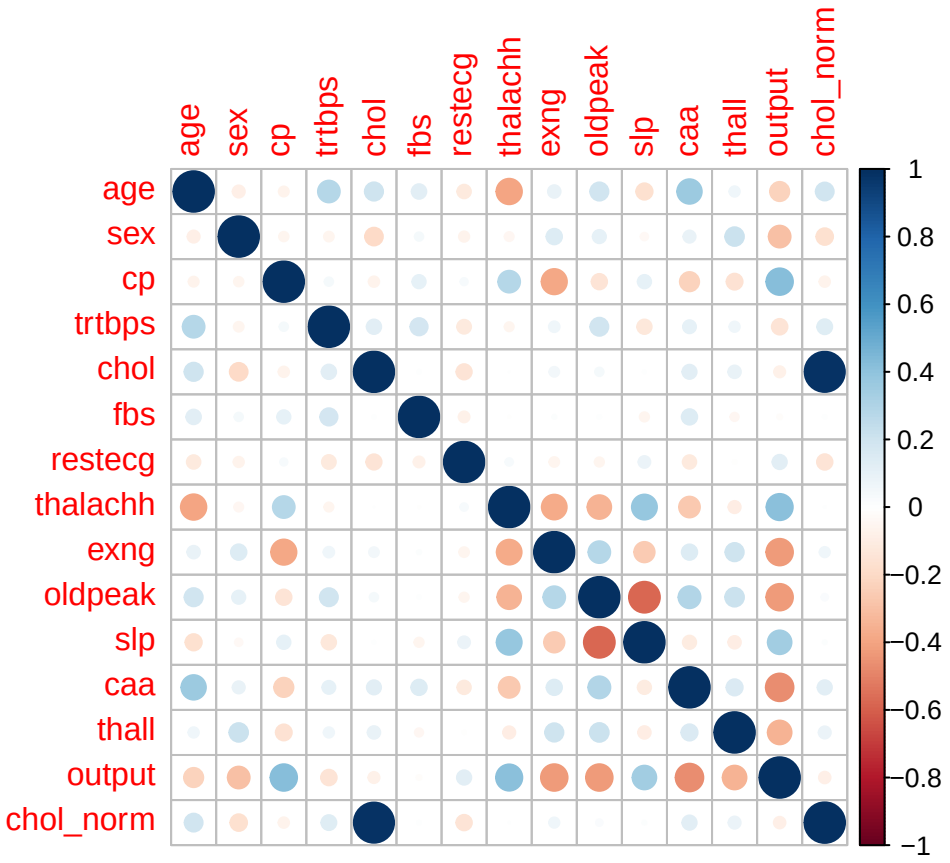
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

Vemos que la variable que hemos creado tiene problemas con la variable original chol por lo que eliminaremos del modelo la variable creada. En la matriz de corrrelación podemos ver mejor esta correlación.

```
# volvemos a porner los valores con formato integer para que poder utilizar corrrplot
heartnum <- heart
heartnum$sex <- as.integer(heart$sex)
heartnum$cp <- as.integer(heart$cp)
heartnum$restecg <- as.integer(heart$restecg)
heartnum$exng <- as.integer(heart$exng)
heartnum$thall <- as.integer(heart$thall)
heartnum$output <- as.integer(heart$output)

corrplot(cor(heartnum), method = "circle")
```



Vamos a crear nuevos modelos con menos variables para ver si conseguimos mejorar el que ya tenemos. Eliminaremos variables que eran poco significativas en el modelo anterior.

```r
modelo2 <- glm(output~. - chol_norm - age - trtbps - restecg - fbs - oldpeak, data = train, family = 'b
summary(modelo2)
```

```
##
## Call:
## glm(formula = output ~ . - chol_norm - age - trtbps - restecg -
##     fbs - oldpeak, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7976  -0.3306   0.1484   0.4680   3.0820
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.533859   2.990508  -0.847 0.396828
## sex1        -1.130799   0.563227  -2.008 0.044673 *
## cp1          1.134181   0.651357   1.741 0.081638 .
## cp2          2.166806   0.562912   3.849 0.000118 ***
## cp3          1.354618   0.654791   2.069 0.038567 *
## chol        -0.004329   0.004214  -1.027 0.304259
## thalachh     0.012961   0.011215   1.156 0.247815
## exng1       -0.818180   0.493795  -1.657 0.097535 .
## slp          1.051927   0.358135   2.937 0.003312 **
## caa         -1.552482   0.292991  -5.299 1.17e-07 ***
## thall1       1.828907   2.594304   0.705 0.480829
## thall2       2.369178   2.501577   0.947 0.343601
## thall3       0.580939   2.498352   0.233 0.816127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 335.48  on 243  degrees of freedom
## Residual deviance: 156.21  on 231  degrees of freedom
## AIC: 182.21
##
## Number of Fisher Scoring iterations: 6
```

Vemos que el valor AIC ha bajado en comparación con el anterior modelo por lo que en teoría este será mejor. Ahora vamos a realizar predicciones para ver la precisión de nuestro modelo.

```r
predict <- predict(modelo2, test, type = 'response')
mat <- table(test$output, predict > 0.5)
mat
```

```
##
##     FALSE TRUE
##   0    21    7
##   1     1   25
```

A continuación vamos a calcular las medidas de sensibilidad y especificidad:

Sensibilidad:

```
mat[4]/(mat[4]+mat[2])
```

```
## [1] 0.9615385
```

Especifidad:

```
mat[1]/(mat[1]+mat[3])
```

```
## [1] 0.75
```

Por último, vamos a ver si la curba roc de nuestro modelo.

```
if (!require("pROC")) install.packages("pROC")
```

```
## Loading required package: pROC
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

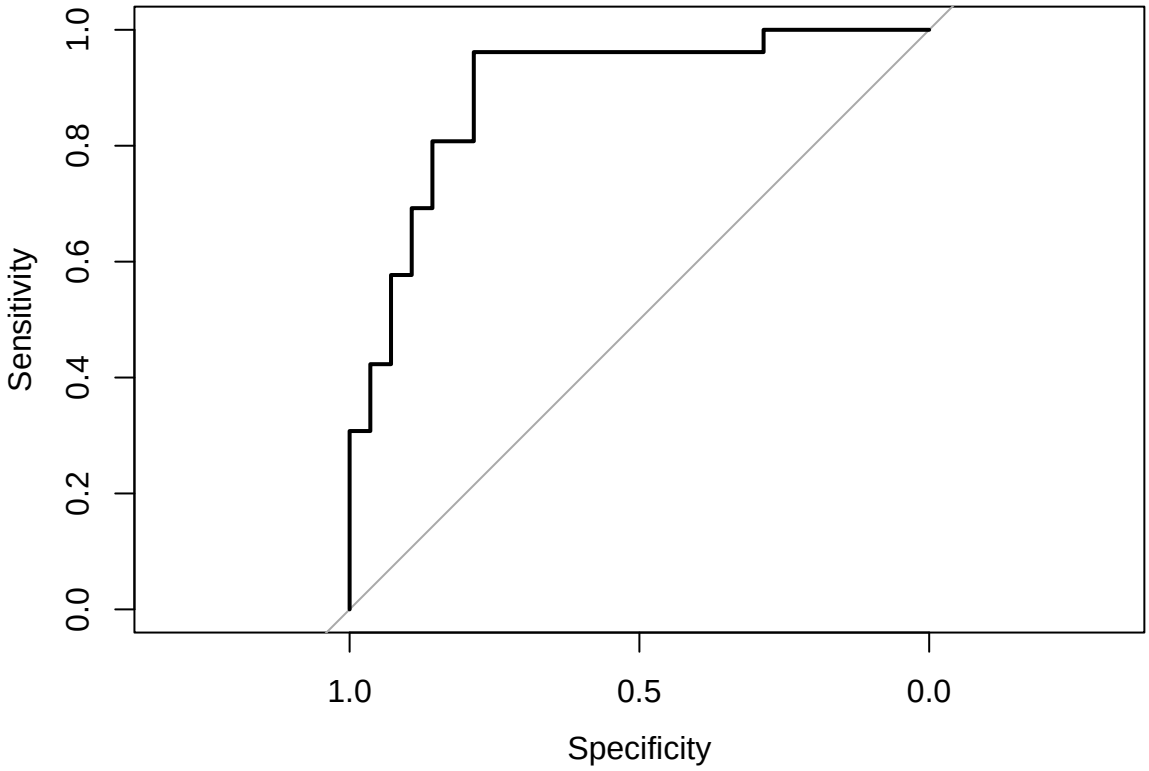
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
#library(pROC)
prob=predict
r=roc(test$output,prob, data=test)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot (r)
```




```
auc(r)
```

```
## Area under the curve: 0.8956
```

Tenemos una curba roc del 89% por lo que tenemos un modelo bastante preciso.

5. Resolución del problema

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

Como se ha visto, se han realizado pruebas estadísticas sobre un conjunto de datos que se correspondía con diferentes variables relativas a pacientes con motivo de cumplir en la medida de lo posible con el objetivo que se planteaba al comienzo. Para cada una de ellas, hemos podido ver cuáles son los resultados que arrojan y qué conocimientos pueden extraerse a partir de ellas.

El modelo de regresión lineal obtenido resulta de utilidad a la hora de realizar predicciones para la variable output dadas unas características concretas. Además también hemos visto como seleccionar variables basandonos en la colinealidad de estas o en la influencia en el modelo.

Antes de realizar estos procesos, se han sometido los datos a un preprocesamiento en el que se ha estudiado los valores nulos y los outliers y los tipos de formato. En el primer caso no se ha encontrado valores nulos. Para el caso del segundo, algunos de los outliers se han dejado como estaban porque no se debian a errores pero en otros casos se han eliminado los registros con valores erroneos. Tambíen se han gestionado los tipos de variable y se han cambiado los que no pertenecian a su tipo original.