Práctica 2: ¿Cómo realizar la limpieza y análisis de datos?

Juan Carlos Escribano Rubio, Juan Javier Corrales Pérez

22 de Diciembre de 2022

Índice

1. Descripción del dataset.

Para realizar la práctica hemos decidido utilizar el dataset propuesto en el enunciado:

https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

El nombre del dataset es "Heart Attack Analysis & Prediction Dataset" y contiene diversas variables con información médica de diversos pacientes y la variable "output" dicotómica que indica si tiene más o menos probabilidad de sufrir un ataque cardíaco.

Este dataset nos permite realizar el análisis de diversos datos médicos que pueden ser relevantes a la predicción de la probabilidad de sufrir un ataque cardíaco. Con estos datos estudiaremos la relevancia que tienen algunas variables básicas como la edad y el sexo y construiremos un modelo para determinar la viabilidad de construir un modelo predictivo.

El detalle de las variables contenidas en el dataset es el siguiente:

- age: Edad del paciente.
- sex: Sexo biológico del paciente (0,1)
- cp: Tipo de dolor torácico, medida en cuatro categorías: (0:Angina típica, 1:Angina atípica, 2:Dolor no anginoso, 3:Asintomático)
- trtbps - Presión arterial en reposo (en mm Hg)
- chol - Colesterol en mg/dl obtenido a través del sensor BMI
- fbs - azúcar en sangre en ayunas > 120 mg/dl, medida en dos categorías: (0:Falso, 1:Verdadero)
- restecg - Resultados electrocardiográficos en reposo, medida en tres categorías: (0:Normal, 1:Normalidad de onda ST-T, 2:Hipertrofia ventricular izquierda)
- thalachh - Frecuencia cardíaca máxima alcanzada.
- exng - Angina inducida por el ejercicio, medida en dos categorías: (0:No, 1:Sí)
- oldpeak - Depresión del segmento ST en prueba de esfuerzo
- slp - Pendiente del segmento ST, medida en tres categorías: (0:Pendiente ascendente, 1:Pendiente plana, 2:Pendiente descendente)
- caa - Número de vasos
- thall - Resultado de la prueba de esfuerzo con talio, medida en cuatro categorías: (0 ~ 3)
- output - variable de destino, medida en dos categorías: (0:Menos posibilidades de ataque al corazón, 1:Más posibilidades de ataque al corazón)

Este dataset nos permite, por un lado, realizar un análisis de cuáles son las variables más relevantes de cara a predecir un ataque cardíaco y por otra modelar un sistema predictivo que nos indique la probabilidad de sufrir un ataque cardíaco en base a las variables medicas de un paciente.

2. Integración y selección

```
# Realizamos la carga de los datos a trabajar.
data_heart <- read.csv("heart.csv", stringsAsFactors = TRUE, header = TRUE, sep=",")

# Mostramos las dimensiones, la estructura y el contenido del data frame cargado.
dim(data_heart)
```

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

```
str(data_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 ...
```

```
summary(data_heart)
```

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

```
# Cambiamos el tipo de datos a factor de algunas variables.
cols<-c("sex","cp","fbs","restecg", "exng", "slp", "thall", "output")
for (i in cols){
data_heart[,i] <- as.factor(data_heart[,i])
}
```

```
# Después de los cambios, analizamos la nueva estructura del conjunto de datos
str(data_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     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
##  $ 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     : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
##  $ 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 ...
```

Dado que todas las variables han sido importadas como tipo **int**, modificamos el tipo de variable a **factor** para todas las variables categóricas de nuestro dataset.

3. Limpieza de los datos.

3.1 ¿Los datos contienen ceros o elementos vacíos?

```
# Estadísticas de valores vacíos
colSums(is.na(data_heart))
```

```
##      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
```

```
# Se comprueba si existen registros duplicados
data_heart[duplicated(data_heart), ]
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
165	38	1	2	138	175	0	1	173	0	0	2	4	2	1

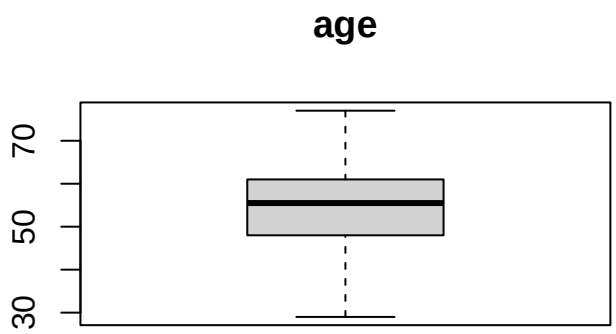
```
# Se eliminan los datos duplicados
data_heart <- data_heart[!duplicated(data_heart), ]
```

Se realiza la comprobación de si existen valores vacíos y se comprueba que no existe ninguno. El dataset tampoco contiene datos a cero, ya que todos los datos importados con este valor son correctos.

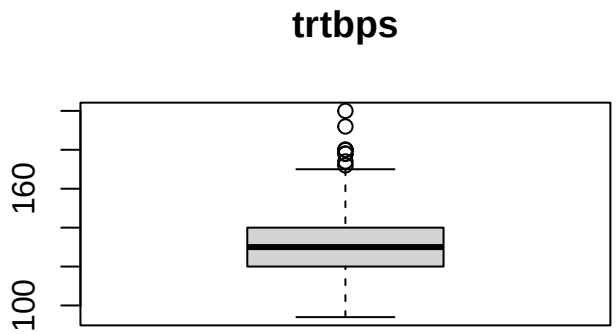
Se comprueba también de si existen datos duplicados. Se detecta de que existe un registro duplicado y se procede a su eliminación del dataset.

3.2 Identifica y gestiona los valores extremos.

```
# Comprobación de age
boxplot(data_heart$age, main="age")
```



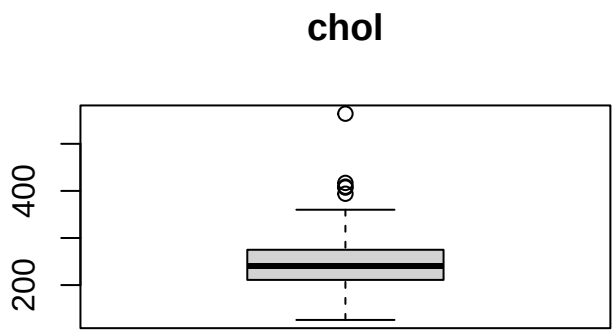

```
# Comprobación de trtbps
boxplot(data_heart$trtbps, main="trtbps")
```




```
x <- boxplot.stats(data_heart$trtbps)$out
idx <- which(data_heart$trtbps %in% x)
sort(data_heart$trtbps[idx])
```

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

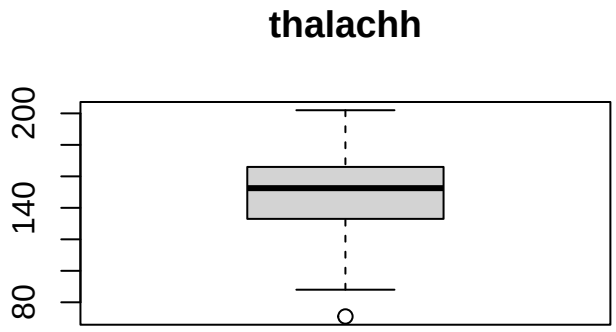
```
# Comprobación de chol
boxplot(data_heart$chol, main="chol")
```




```
x <- boxplot.stats(data_heart$chol)$out
idx <- which(data_heart$chol %in% x)
sort(data_heart$chol[idx])
```

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

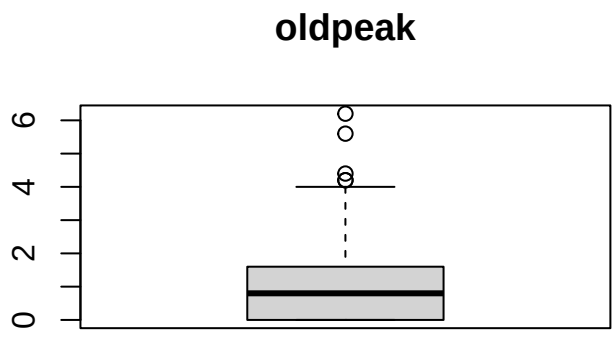
```
# Comprobación de thalachh
boxplot(data_heart$thalachh, main="thalachh")
```




```
x <- boxplot.stats(data_heart$thalachh)$out
idx <- which(data_heart$thalachh %in% x)
sort(data_heart$thalachh[idx])
```

```
## [1] 71
```

```
# Comprobación de oldpeak
boxplot(data_heart$oldpeak, main="oldpeak")
```




```
x <- boxplot.stats(data_heart$oldpeak)$out
idx <- which(data_heart$oldpeak %in% x)
sort(data_heart$oldpeak[idx])
```

```
## [1] 4.2 4.2 4.4 5.6 6.2
```

De las variables analizadas, solo se detecta que existen valores extremos en la variable **chol**. En el resto de variables, a pesar de que existen valores atípicos, estos no se consideran valores anómalos.

```
# chol
# Se asigna a NA los valores > 500. El resto se deja igual.
data_heart$chol[data_heart$chol > 500 ] <- NA

#Check
sum(is.na(data_heart$chol))
```

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

```
# Se calcula la media aritmética por género
idx <- which(is.na(data_heart$chol))
mean.f <- round(mean(data_heart$chol[data_heart$sex == 0], na.rm=TRUE ))
mean.m <- round(mean(data_heart$chol[data_heart$sex == 1], na.rm=TRUE ))

# Se asignan los nuevos valores
data_heart$chol[idx] <- ifelse(data_heart$sex[idx] == 0, mean.f, mean.m)
data_heart$chol[idx]
```

```
## [1] 258
```

```
# Exportamos nuestro dataset limpio a CSV
write.csv(data_heart, "heart_procesado.csv")
```

Se corrigen los valores extremos de la variable **chol** aplicando la imputación de su nuevo valor por la media aritmética de los registros del mismo género, es decir, separado por género.

4. Análisis de los datos.

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

A continuación indico las preguntas que se quieren responder, las variables que utilizaremos para ello y el tipo de análisis que se realizará en cada caso:

- ¿Tiene influencia el sexo en la probabilidad de tener un ataque cardíaco?
  - Variables analizadas:
    - sex, output
  - Tipo de análisis
    - Indendencia de dos variables categóricas
      - Tabla de contingencia
      - Chi cuadrado

- ¿La edad de las personas con más probabilidad de ataque cardíaco es significativamente diferente de las que tienen menos probabilidad?
  - Variables analizadas:
    - age, output
  - Tipo de análisis
    - Tabla de contingencia
    - Chi cuadrado

- Creación de un modelo de regresión logística
  - Variables analizadas:
    - output -> age, sex, cp, trtbps, thalachh, exng, oldpeak, caa
  - Tipo de análisis:
    - Correlación entras las variables independientes
    - Creación del modelo de regresión
    - Evaluación de la calidad del modelo

4.2 Comprobación de la normalidad y homogeneidad de la varianza.

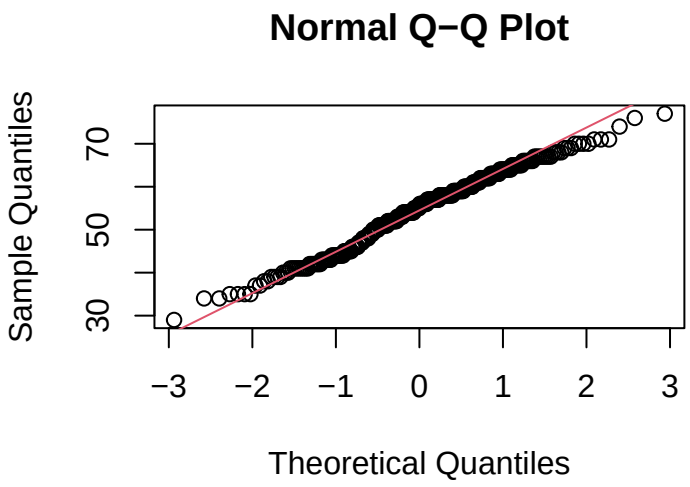
4.2.1 Comprobación de la normalidad

A continuación comprobaremos la normalidad de las variables que vamos a analizar.

– **Age**

- Un qqplot donde observamos que la curva se ajusta bastante bien a la normal.
- Un test de Kolmogorov-Smirnov donde obtenemos un p-value > 0.05 por lo que podemos considerar la variable como normal.

```
qqnorm(data_heart$age)
qqline(data_heart$age, col=2)
```




```
ks.test(data_heart$age, pnorm, mean(data_heart$age), sd(data_heart$age))
```

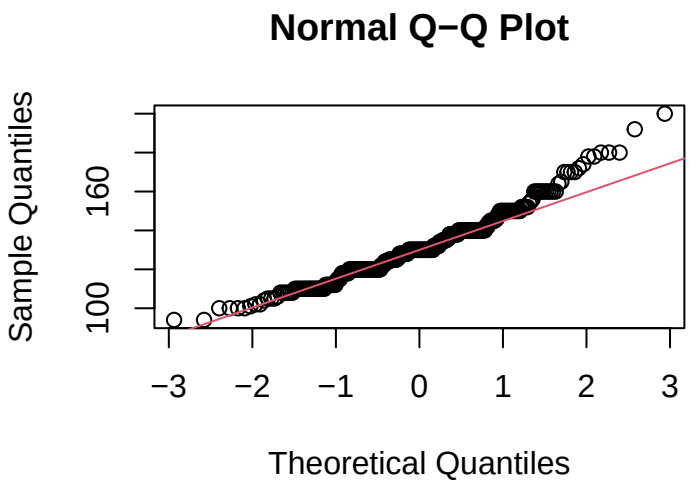
```
## Warning in ks.test.default(data_heart$age, pnorm, mean(data_heart$age), : ties
## should not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  data_heart$age
## D = 0.075788, p-value = 0.06228
## alternative hypothesis: two-sided
```

– **trtbps**

- Un qqplot donde observamos que la curva no se ajusta a la normal.
- Un test de Kolmogorov-Smirnov donde obtenemos un p-value < 0.05 por lo que podemos considerar la variable como no normal.
- Un test de Shapiro-Wilk donde obtenemos un p-value < 0.05 por lo que podemos considerar la variable como no normal.

```
qqnorm(data_heart$trtbps)
qqline(data_heart$trtbps, col=2)
```




```
ks.test(data_heart$trtbps, pnorm, mean(data_heart$trtbps), sd(data_heart$trtbps))
```

```
## Warning in ks.test.default(data_heart$trtbps, pnorm, mean(data_heart$trtbps), :
## ties should not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  data_heart$trtbps
## D = 0.10258, p-value = 0.003475
## alternative hypothesis: two-sided
```

```
shapiro.test(data_heart$trtbps)
```

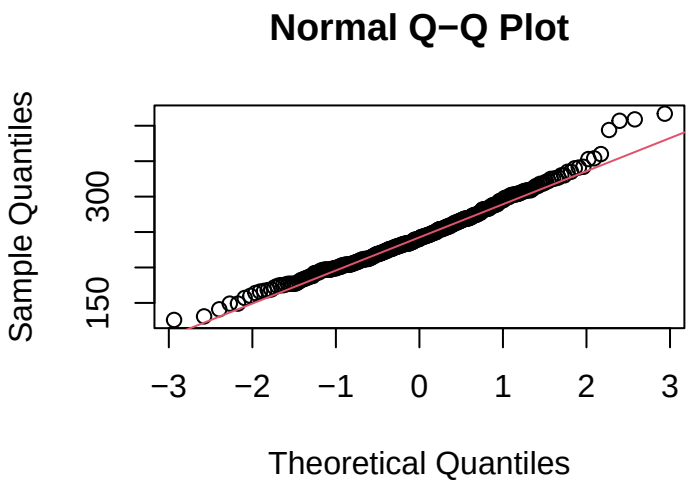
```
##
##  Shapiro-Wilk normality test
##
## data:  data_heart$trtbps
## W = 0.96573, p-value = 1.419e-06
```

– **chol**

- Un qqplot donde observamos que la curva se ajusta bastante bien a la normal.
- Un test de Kolmogorov-Smirnov donde obtenemos un p-value > 0.05 por lo que podemos considerar la variable como normal.
- Un test de Shapiro-Wilk donde obtenemos un p-value < 0.05 por lo que podemos considerar la variable como no normal.

Dado que tanto en el qqplot como en el test de Kolmogorov-Smirnov se observa normalidad consideraremos la variable como normal.

```
qqnorm(data_heart$chol)
qqline(data_heart$chol, col=2)
```




```
ks.test(data_heart$chol, pnorm, mean(data_heart$chol), sd(data_heart$chol))
```

```
## Warning in ks.test.default(data_heart$chol, pnorm, mean(data_heart$chol), : ties
## should not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  data_heart$chol
## D = 0.04789, p-value = 0.4927
## alternative hypothesis: two-sided
```

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

```
##
##  Shapiro-Wilk normality test
##
## data:  data_heart$chol
## W = 0.98292, p-value = 0.001154
```

4.2.2 Comprobación de la homogeneidad de la varianza

– **Age**

No existe homogeneidad de la varianza entre edad y output (p-value < 0.05).

```
leveneTest(age ~ output, data=data_heart)
```

	Df	F value	Pr(>F)
group	1	7.634937	0.0060785
	300	NA	NA

– **trtbps**

Existe homogeneidad de la varianza entre trtbps y output (p-value > 0.05).

```
leveneTest(trtbps ~ output, data=data_heart)
```

	Df	F value	Pr(>F)
group	1	1.791063	0.1818101
	300	NA	NA

– **chol**

Existe homogeneidad de la varianza entre chol y output (p-value > 0.05).

```
leveneTest(chol ~ output, data=data_heart)
```

	Df	F value	Pr(>F)
group	1	0.8635136	0.3535044
	300	NA	NA

– **sex**

No existe homogeneidad de la varianza entre sex y output (p-value < 0.05).

```
data_heart$sex_num <- as.numeric(data_heart$sex)
leveneTest(sex_num ~ output, data=data_heart)
```

	Df	F value	Pr(>F)
group	1	26.24095	5e-07
	300	NA	NA

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.

4.3.1 ¿Tiene influencia el sexo en la probabilidad de tener un ataque cardíaco?

Para responder a esta pregunta realizaremos un test de independencia para dos variables categóricas.

Como primer paso calculamos la tabla de contingencia y representamos sus valores en un gráfico de barras.

Tanto en la tabla como en él gráfico observamos diferencias en los valores para cada sexo y que parece existir alguna relación entre ambas, para comprobar si estas diferencias son estadísticamente significativas realizamos la prueba de chi cuadrado.

Como resultado del test chi-cuadrado obtenemos un p-value<0.05 lo que significa que existe una diferencia significativa en la distribución entre los sexos con respecto a la probabilidad de tener un ataque cardíaco.

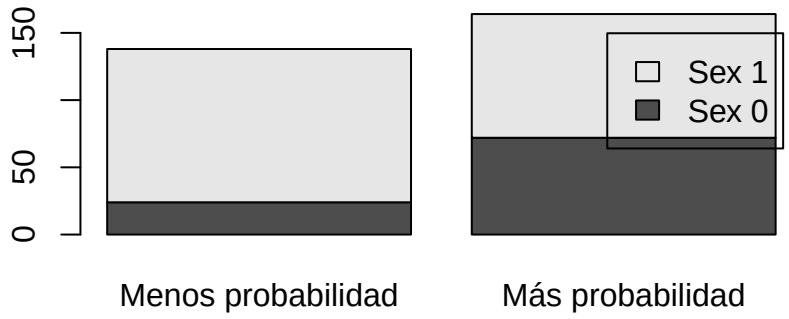
```
tabla_contingencia <- table(data_heart$sex, data_heart$output)

table(data_heart$sex[data_heart$output == 1])
```

```
##
##  0  1
## 72 92
```

```
rownames(tabla_contingencia) <- c("Sex 0","Sex 1")
colnames(tabla_contingencia) <- c("Menos probabilidad","Más probabilidad")
print(tabla_contingencia)
```

```
##
##         Menos probabilidad Más probabilidad
##   Sex 0                 24               72
##   Sex 1                114               92
```

```
barplot(tabla_contingencia, legend = TRUE)
```




```
chisq.test(tabla_contingencia)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tabla_contingencia
## X-squared = 23.084, df = 1, p-value = 1.551e-06
```

```
nrow(subset(data_heart, sex == 1 & output == 0))
```

```
## [1] 114
```

4.3.2 ¿La edad de las personas con más probabilidad de ataque cardíaco es significativamente diferente de las que tienen menos probabilidad?

Para responder a esta pregunta realizaremos un contraste de hipótesis para comprobar si la media de edad en el grupo con más probabilidad de ataque cardíaco es diferente del grupo con menos probabilidad.

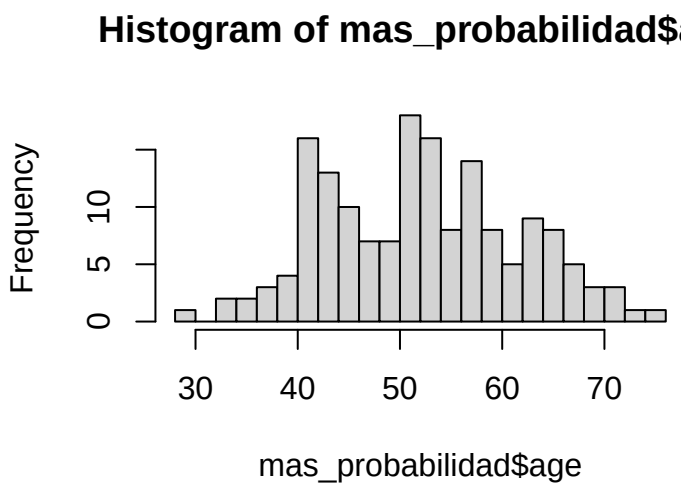
Como primer paso, se incluyen un histograma para la población de la muestra con más probabilidad de ataque cardíaco y otra con la que tiene menos a efectos de observar las diferencias en la distribución de edades. Observando estos histogramos observamos que la distribución es diferente.

Para realizar el contraste de hipotesis y ya que no existe homogeneidad en la varianza entre age y output, se tiene que pasar el parámetro var.equal = False a la función t.test (así utilizará el t-test de Welch) para que lo

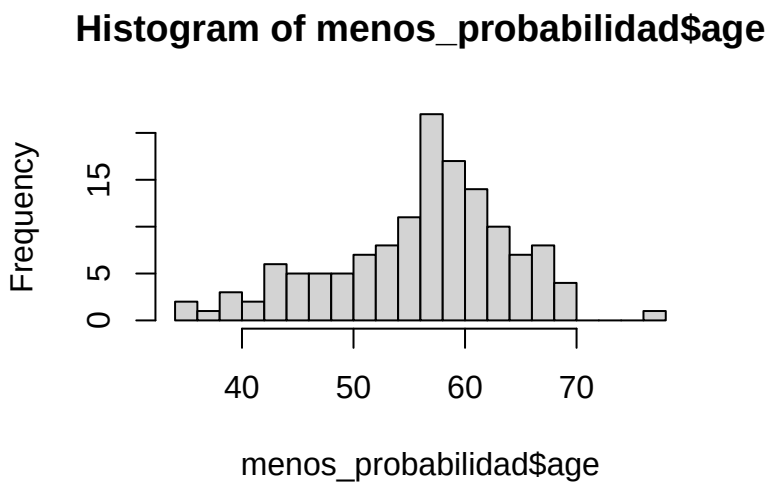
tenga en cuenta.

El resultado del test indica que es muy poco probable que la diferencia observada en la media de la edad entre los dos grupos sea debido al azar. Por lo tanto, se puede concluir que hay una diferencia significativa en la edad entre los dos grupos.

```
mas_probabilidad <- subset(data_heart, output == 1)
menos_probabilidad <- subset(data_heart, output == 0)
hist(mas_probabilidad$age,breaks=20)
```




```
hist(menos_probabilidad$age,breaks=20)
```




```
t.test(data_heart$age ~ data_heart$output,var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  data_heart$age by data_heart$output
## t = 3.994, df = 299.99, p-value = 8.177e-05
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  2.037315 5.994852
## sample estimates:
## mean in group 0 mean in group 1
##        56.60145        52.58537
```

4.3.3 Creación de un modelo de regresión logística

```
# Generamos los datos de entrenamiento y test para el modelo
set.seed(123)
ind <- sample(seq_len(nrow(data_heart)), size = round(.8 * dim(data_heart)[1]))
training <- data_heart[ind, ]
testing <- data_heart[-ind, ]

# Estimamos el modelo
model.logist1=glm(formula=output~age+sex+cp+trtbps+thalachh+exng+oldpeak+caa,data=training, family=bino
summary(model.logist1)
```

```
##
## Call:
## glm(formula = output ~ age + sex + cp + trtbps + thalachh + exng +
##     oldpeak + caa, family = binomial, data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3443  -0.4292   0.1697   0.5618   2.4133
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.861531   2.644391   0.326 0.744579
## age         -0.008444   0.025302  -0.334 0.738572
## sex1        -1.666856   0.466662  -3.572 0.000354 ***
## cp1          1.529536   0.689276   2.219 0.026484 *
## cp2          1.439033   0.480508   2.995 0.002746 **
## cp3          1.930092   0.670772   2.877 0.004009 **
## trtbps      -0.020525   0.011759  -1.745 0.080911 .
## thalachh     0.030669   0.010972   2.795 0.005188 **
## exng1       -1.224934   0.436223  -2.808 0.004984 **
## oldpeak     -0.649466   0.206321  -3.148 0.001645 **
## caa         -0.748527   0.197915  -3.782 0.000156 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 333.48  on 241  degrees of freedom
## Residual deviance: 172.15  on 231  degrees of freedom
## AIC: 194.15
##
## Number of Fisher Scoring iterations: 5
```

Observamos que la variable **age** no es significativa, por lo que procedemos a eliminarla del modelo.

```r
model.logist2=glm(formula=output~sex+cp+trtbps+thalachh+exng+oldpeak+caa,data=training, family=binomial)
summary(model.logist2)
```

```
##
## Call:
## glm(formula = output ~ sex + cp + trtbps + thalachh + exng +
##     oldpeak + caa, family = binomial, data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3323  -0.4302   0.1754   0.5742   2.4025
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.33737    2.12091   0.159 0.873616
## sex1        -1.64790    0.46319  -3.558 0.000374 ***
## cp1          1.52029    0.68807   2.209 0.027141 *
## cp2          1.44014    0.48033   2.998 0.002716 **
## cp3          1.92808    0.67121   2.873 0.004072 **
## trtbps      -0.02165    0.01128  -1.918 0.055067 .
## thalachh     0.03197    0.01026   3.118 0.001822 **
## exng1       -1.21390    0.43418  -2.796 0.005177 **
## oldpeak     -0.65016    0.20680  -3.144 0.001667 **
## caa         -0.76042    0.19538  -3.892 9.94e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 333.48  on 241  degrees of freedom
## Residual deviance: 172.26  on 232  degrees of freedom
## AIC: 192.26
##
## Number of Fisher Scoring iterations: 5
```

```r
vif(model.logist2)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## sex      1.144055  1        1.069605
## cp       1.346740  3        1.050866
## trtbps   1.080604  1        1.039521
## thalachh 1.087514  1        1.042840
## exng     1.139122  1        1.067297
## oldpeak  1.169980  1        1.081656
## caa      1.050561  1        1.024969
```

Vemos que ahora todas las variables son significativas y que no existe colinealidad.

```r
pred_test <- predict(object = model.logist2, newdata = testing, type = "response")

## Predicción
testing$prediction <- ifelse(pred_test < 0.5 ,0, 1)
prediction <- as.factor(testing$prediction)
```

```
true <- testing$output
glimpse(testing[,c(14,16)])
```

```
## Rows: 60
## Columns: 2
## $ output     <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ prediction <dbl> 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,~
```

```
# Matriz confusión
confusionMatrix(prediction, true, positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 16  2
##          1 12 30
##
##                Accuracy : 0.7667
##                  95% CI : (0.6396, 0.8662)
##     No Information Rate : 0.5333
##     P-Value [Acc > NIR] : 0.0001655
##
##                   Kappa : 0.5205
##
##  Mcnemar's Test P-Value : 0.0161569
##
##             Sensitivity : 0.9375
##             Specificity : 0.5714
##          Pos Pred Value : 0.7143
##          Neg Pred Value : 0.8889
##              Prevalence : 0.5333
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.7000
##       Balanced Accuracy : 0.7545
##
##        'Positive' Class : 1
##
```

Una de las métricas que se pueden usar para evaluar el modelo es la **exactitud** (accuracy), que es la proporción entre las predicciones correctas hechas por el modelo y el total de predicciones. En nuestro caso se obtiene un valor de 0.7667 con un intervalo de confianza de (0.6396, 0.8662).

Por otro lado:

- La **sensibilidad** (sensitivity): 0.9375. Proporción de casos positivos correctamente clasificados.
- La **especificidad** (specificity): 0.5714. Proporción de casos negativos correctamente clasificados.

A la vista de estos resultados, se puede concluir que es un buen modelo.

```
# Realizamos el test de Chi-cuadrado
dev <- model.logist2$deviance
nullDev <- model.logist2$null.deviance
Chi_Obs <- nullDev - dev
Chi_Obs
```

```
## [1] 161.2185
```

```
# Calculamos la probabilidad asociada al estadístico del contraste
gl <- model.logist2$df.null - model.logist2$df.residual
chi_prob <- 1 - pchisq(Chi_Obs,gl)
chi_prob
```

```
## [1] 0
```

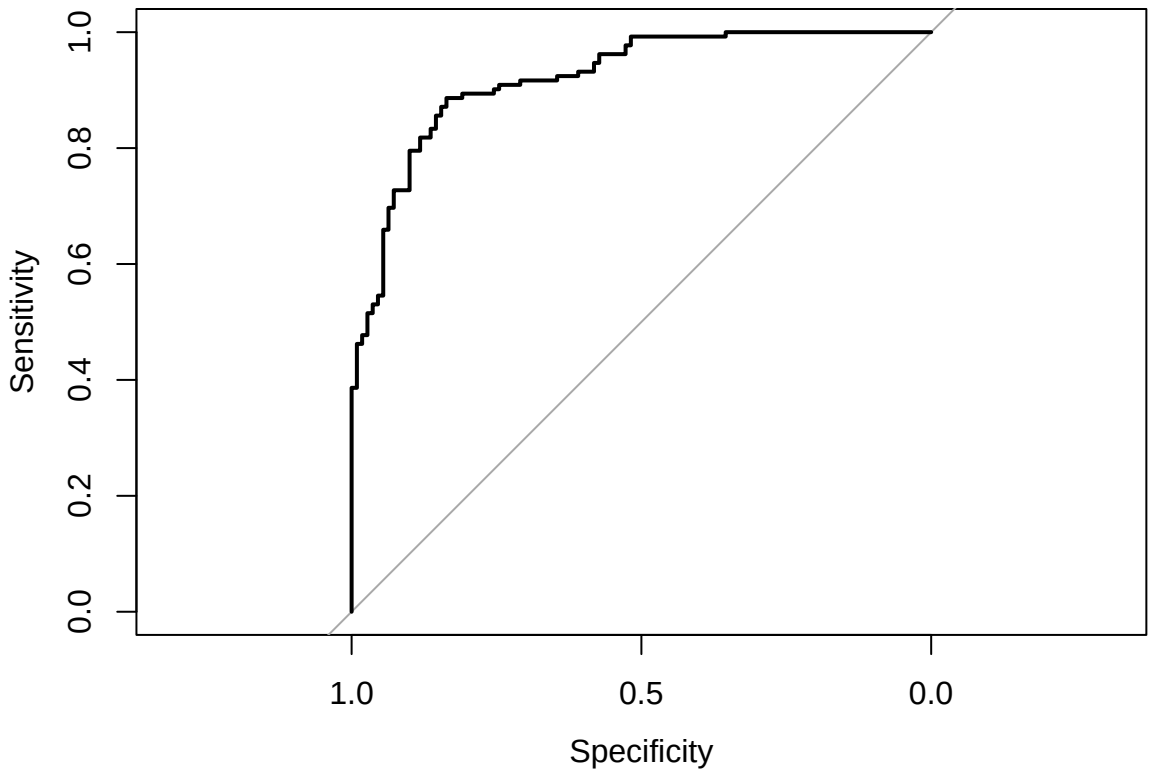
A la vista de los resultados el ajuste es bueno con un p-value de 0.

```
# Se realiza el dibujo de la curva ROC
prob_low = predict(model.logist2, training, type="response")
r = roc(training$output, prob_low, data=training)
```

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

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

```
plot(r)
```




```
value.auc <- auc(r)
value.auc
```

```
## Area under the curve: 0.921
```

El área por debajo de esa curva toma el valor de 0.921, por lo que la habilidad del modelo para predecir es muy buena.

6. Resolución del problema.

Dado el problema que planteabamos consistente en por un lado, analizar algunas de las variables de cara a determinar si tienen influencia en la probabilidad y por otro crear un modelo predictivo con el fin de evaluar su calidad, hemos llegado a las siguientes conclusiones:

- Existen diferencias significativas entre la distribución de sexos en la probabilidad de sufrir un ataque cardíaco
- Existen diferencias significativas entre las medias de edad de las personas con más y menos probabilidad de sufrir un ataque cardíaco.
- Es posible obtener un modelo de regresión logística que obtenga unos buenos resultados predictivos (Sensibilidad = 0.94, especificidad = 0.57)

Como conclusión del análisis podemos indicar que con el conjunto de datos de entrada que hemos utilizado es viable la creación de modelos predictivos que permiten obtener una estimación de la probabilidad de sufrir un ataque cardíaco del paciente en base a sus datos médicos.

Tabla de contribuciones

Contribuciones	Firma
Investigación previa	jcescribano, jcorralesp
Redaccción de las respuestas	jcescribano, jcorralesp
Desarrollo del código	jcescribano, jcorralesp
Parcipación en el vídeo	jcescribano, jcorralesp