Práctica 2: Proyecto analítico. Integración, limpieza, validación y análisis

Diego García García - Alba Caderno Fernández

28 de diciembre de 2022

Índice

1 Detalles de la actividad

1.1 Descripción

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

Un ejemplo de dataset con el que podéis trabajar es el "Heart Attack Analysis & Prediction dataset": https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-predictiondataset

Importante: si se elige un dataset diferente al propuesto es importante que este contenga una amplia variedad de datos numéricos y categóricos para poder realizar un análisis más rico y poder responder a las diferentes preguntas planteadas en el enunciado de la práctica.

1.2 Objetivos

Los objetivos concretos de esta práctica son:

Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.

Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.

Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.

Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.

Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.

Desarrollar las habilidades de aprendizaje que les permitan continuarestudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.

Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

1.3 Competencias

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

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

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

2 Resolución

2.1 Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El Dataset escogido en el enunciado de la práctica 2 nos muestra los datos para la clasificación de ataques al corazón. Dentro del conjunto de datos nos encontramos con 14 atributos provenientes de 4 bases de datos distintas del Instituto Húngaro de Cardiología en Budapest por Andras Janosi, el Hospital Universitario, Zurich, en Suiza por William Steinbrunn, el Hospital Universitario, Basilea, en Suiza por Matthias Pfisterer y el V.A. Medical Center en Long Beach y Cleveland Clinic Foundation por Robert Detrano.

El atributo objetivo(output) hace referencia a la presencia de enfermedad cardíaca en el paciente siendo su escala de valores de 0 a 4. Por tanto, a raíz de los valores del conjuntos de datos se quiere averiguar la influencia de las distintas variables con el fin de determinar la presencia (¿es propenso?) de enfermedad cadíaca en los pacientes de estudio.

Nota: Tener en cuenta que en el atributo thall para el valor 0 se ha rellenado el dato con el value null por lo que puede ser necesario usar alguna técnica de ML supervisada para asignarle un valor “aproximado”.

Age: Edad del paciente en años.

Sex: Sexo del paciente.Valor 1: masculino, Valor 0: femenino

cp: Tipo de dolor en el pecho.Valor 0: Asintomático, Valor 1: Angina típica,Valor 2: Angina atípica,Valor 3: Dolor no-anginoso

trtbps: Presión arterial en reposo medida en el servicio de admisión del hospital en mm Hg. La presión sistólica.

chol: Colesterol en mg/dl obtenido a través del sensor BMI

fbs: Glucemia en ayunas $>$ 120 mg/dl. Valor 1: Verdadero,Valor 0: Falso

rest_ecg: Resultados electrocardiográficos en reposo. Valor 0: Normal,Valor 1: Tener anomalías en la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST $>$ 0,05 mV),Valor 2: Mostrar hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes.

thalach: Frecuencia cardíaca máxima alcanzada.

exang: Angina inducida por el ejercicio.Valor 1: Si, Valor 0: No

old peak: Depresión del ST inducida por el ejercicio en relación con el reposo.

slp: La pendiente del segmento ST de ejercicio máximo (El segmento ST representa el período isoeléctrico cuando los ventrículos se encuentran entre la despolarización y la repolarización (https://es.wikipedia.org/wiki/Segmento_ST)).Valor 0: Pendiente abajo,Valor 1: Plana,Valor 2: Pendiente arriba

caa: Número de vasos principales (0-3) coloreados por fluoroscopia

thall: thalassemia (Resultado de la prueba de esfuerzo con talio). Valor 0: null,Valor 1: Defecto fijo,Valor 2: Normal,Valor 3: Defecto reversible

output: Diagnóstico de enfermedad cardíaca (estado de enfermedad angiográfico). Si el valor es 1 entonces el paciente es más propenso a tener una enfermedad cardíaca mientras que si es 0 entonces la situación del paciente no es crítica. El estrechamiento del vaso es peligroso ya que es posible que la sangre no bombee a través del corazón, lo que puede provocar un paro cardíaco.Valor 0: $< 50\%$ estrechamiento del diámetro. Menos posibilidades de ataque al corazón,Valor 1: $> 50\%$ estrechamiento del diámetro. Más posibilidades de ataque al corazón.

2.2 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.

```
# Carga del fichero de datos y lectura de los mismos
dt_heart <- read.csv("heart.csv", header = TRUE)
```

```
# Visualización previa de los datos.
# Tipo de dato asignado a cada campo para hacernos una idea del conjunto.
str(dt_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 ...
```

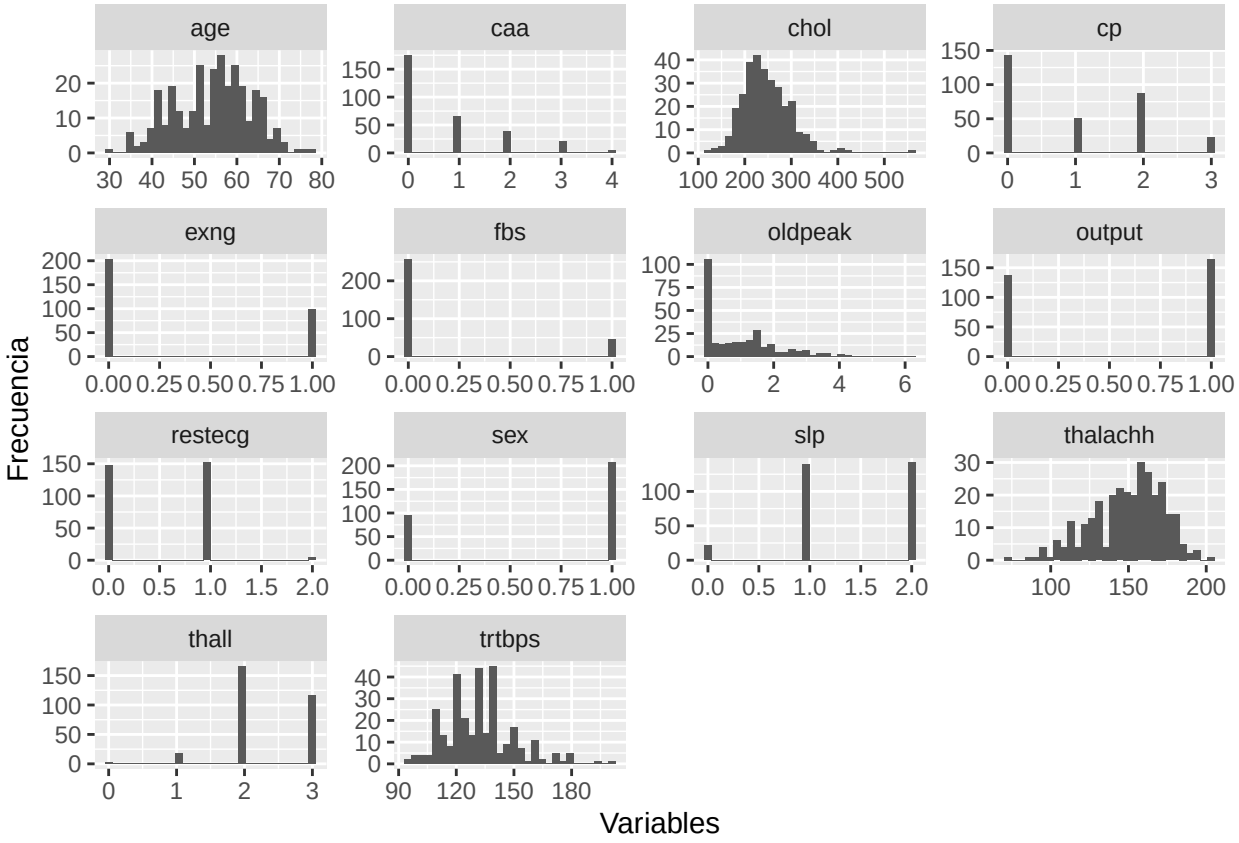
```
sapply(dt_heart, function(x) class(x))
```

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

```
#install.packages("ggplot2")
library(purrr)
library(tidyr)
library(ggplot2)
```

```
# Se realiza un estudio visual de todas las variables para analizar sus distribuciones así como posibles
dt_heart %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    labs(x="Variables",y= "Frecuencia")  +
    geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



En base a los resultados obtenidos, del conjunto de datos, podemos obtener una relación de primeras ideas como: La asignación automática de tipo de datos de R al conjunto se basa en valores enteros y numéricos.Los pacientes de estudio se sitúan en edades comprendidas entre los 29 y los 77 años diendo la media de 54 años.Existe más cantidad de personas con sexo masculino que con femenino.Una minoría de las personas de estudio presenta dolor no-anginoso frente a la mayoría que son asintomáticos.La presión sistólica de los pacientes se centra en torno a los 130mm Hg.La distribución de los datos de la variable colesterol presenta cierta cola a la derecha por lo que los datos se concentrando en torno a los 250 mg/dl.La mayoría de los pacientes no presenta glucemia en ayunas por encima de los 120 mg/dl.La mayoría de los pacientes no presenta angina inducida por el ejercicio.

2.3 Limpieza de los datos.

2.3.1 ¿Los datos contienen ceros o elementos vacíos? Gestiona cada uno de estos casos.

```
# Verificar que no existen elementos NA.
sapply(dt_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
```

```
# Verificar en base a la descripción de los atributos los valores distintos de cada uno de ellos para d
head(sapply(dt_heart, function(x) unique(x)))
```

```
## $age
##  [1] 63 37 41 56 57 44 52 54 48 49 64 58 50 66 43 69 59 42 61 40 71 51 65 53 46
## [26] 45 39 47 62 34 35 29 55 60 67 68 74 76 70 38 77
##
## $sex
## [1] 1 0
##
## $cp
## [1] 3 2 1 0
##
## $trtbps
##  [1] 145 130 120 140 172 150 110 135 160 105 125 142 155 104 138 128 108 134 122
## [20] 115 118 100 124  94 112 102 152 101 132 148 178 129 180 136 126 106 156 170
## [39] 146 117 200 165 174 192 144 123 154 114 164
##
## $chol
##   [1] 233 250 204 236 354 192 294 263 199 168 239 275 266 211 283 219 340 226
##  [19] 247 234 243 302 212 175 417 197 198 177 273 213 304 232 269 360 308 245
##  [37] 208 264 321 325 235 257 216 256 231 141 252 201 222 260 182 303 265 309
##  [55] 186 203 183 220 209 258 227 261 221 205 240 318 298 564 277 214 248 255
##  [73] 207 223 288 160 394 315 246 244 270 195 196 254 126 313 262 215 193 271
##  [91] 268 267 210 295 306 178 242 180 228 149 278 253 342 157 286 229 284 224
## [109] 206 167 230 335 276 353 225 330 290 172 305 188 282 185 326 274 164 307
## [127] 249 341 407 217 174 281 289 322 299 300 293 184 409 259 200 327 237 218
## [145] 319 166 311 169 187 176 241 131
##
## $fbs
## [1] 1 0
```

Como se aprecia en los resultados obtenidos será necesario tratar el atributo:

thall: ya que en un apartado anterior, en el que se describe el dataset, se ha puntualizado que existen valores 0 que hacen referencia a un valor "null" y en concreto son dos observaciones. Por tanto, para dar sentido a estos valores se usará un método de imputación de valores basado en la similitud o diferencia entre los registros. Se usará kNN imputation ya que damos por hecho que los registros están relacionados. Igualmente siempre será mejor tratar los resultados de análisis de datos cercanos frente a datos vacíos.

caa: se aprecia un valor fuera del rango definido. Dicho valor es el 4 que se tratará como un NA y se procederá de la misma manera que para el atributo anterior.

```
suppressWarnings(suppressMessages(library(VIM)))
dt_heart$thall[dt_heart$thall == 0] <- NA # Sustituir valor 0(null) por NA
dt_heart$caa[dt_heart$caa == 4] <- NA # Sustituir valor 0(null) por NA
dt_heart$thall <- kNN(dt_heart)$thall
dt_heart$caa <- kNN(dt_heart)$caa
unique(dt_heart$thall)
```

```
## [1] 1 2 3
```

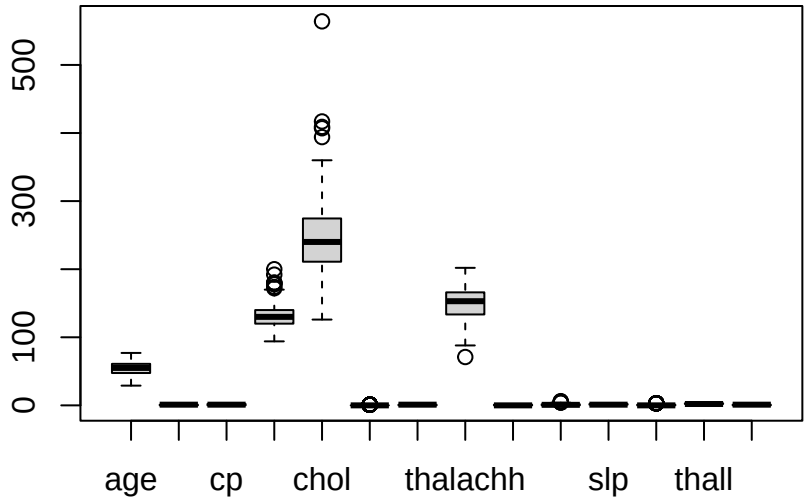
```
unique(dt_heart$caa)
```

```
## [1] 0 2 1 3
```

2.3.2 Identifica y gestiona los valores extremos.

Para poder visualizar de una manera rápida y sencilla para el humano realizamos un boxplot o gráfico de caja.

```
boxplot(dt_heart)
```




```
print("Val. extr. trtbps")
```

```
## [1] "Val. extr. trtbps"
```

```
boxplot.stats(dt_heart$trtbps)$out
```

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

```
print("Val. extr. chol")
```

```
## [1] "Val. extr. chol"
```

```
boxplot.stats(dt_heart$chol)$out
```

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

```
print("Val. extr. thalach")
```

```
## [1] "Val. extr. thalach"
```

```
boxplot.stats(dt_heart$thalach)$out
```

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

```
print("Val. extr. oldpeak")
```

```
## [1] "Val. extr. oldpeak"
```

```
boxplot.stats(dt_heart$oldpeak)$out
```

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

```
print("Val. extr. caa")
```

```
## [1] "Val. extr. caa"
```

```
boxplot.stats(dt_heart$caa)$out
```

```
##  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

En base a los resultados del gráfico se identifican los siguientes atributos con posibles valores extremos:

trtbps - chol - thalach - exang - oldpeak - caa

Para el análisis de estudio no se tratarán estos valores como extremos ya que a pesar de ser cierto que se alejan de la media, son valores que se consideran reales y deben ser tratados en el estudio. Eliminarlos provocaría distorsión en la interpretación de los resultados.

2.4 Análisis de los datos.

2.4.1 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?)

Atendiendo a la definición, expuesta en el primer apartado, de cada variables del conjunto de datos, los agruparemos de la siguiente manera:

Grupo 1: Con el fin de estudiar el impacto de la glucemia en ayunas se seleccionarán aquellos datos cuyos pacientes presenten un nivel de Glucemia por encima de 120 mg/dl. La finalidad de esta prueba estadística es realizar un contraste de hipótesis para averiguar si los índices de Glucemia indican más inclinación a la enfermedad cardíaca. Esta variable nos permitirá identificar pacientes propensos a la enfermedad con valores de un indicador "pre-enfermedad cardíaca", es decir, podremos obtener resultados de un indicador sin que el paciente, posiblemente, haya sufrido una enfermedad cardíaca aún. De ahí la importancia de estudiar esta variable y su posible predicción.

Grupo 2: Con el fin de estudiar el impacto de la pendiente del segmento ST de ejercicio máximo se seleccionarán aquellos datos cuyos pacientes presenten una pendiente arriba. La finalidad de esta prueba estadística es realizar un contraste de hipótesis para averiguar si el hecho de tener la pendiente del segmento ST de ejercicio máximo hacia arriba indica más inclinación a la enfermedad cardíaca.

Indistintamente de la creación de los grupos será muy interesante realizar un análisis de la correlación entre las variables del conjunto de datos para poder obtener más información sobre su comportamiento.

```
# Grupo1
dt_heart_glucemia <- dt_heart[dt_heart$fbs == 1, ]
str(dt_heart_glucemia)
```

```
## 'data.frame':    45 obs. of  14 variables:
##  $ age     : int  63 52 58 61 59 65 53 54 71 58 ...
##  $ sex     : int  1 1 0 1 1 0 1 0 0 1 ...
##  $ cp      : int  3 2 3 2 2 2 2 2 2 2 ...
##  $ trtbps  : int  145 172 150 150 150 140 130 135 110 140 ...
##  $ chol    : int  233 199 283 243 212 417 197 304 265 211 ...
##  $ fbs     : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ restecg : int  0 1 0 1 1 0 0 1 0 0 ...
##  $ thalachh: int  150 162 162 137 157 157 152 170 130 165 ...
##  $ exng    : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 0.5 1 1 1.6 0.8 1.2 0 0 0 ...
##  $ slp     : int  0 2 2 1 2 2 0 2 2 2 ...
##  $ caa     : int  0 0 0 0 0 1 0 0 1 0 ...
##  $ thall   : int  1 3 2 2 2 2 2 2 2 2 ...
##  $ output  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
# Grupo2
dt_heart_pendiente <- dt_heart[dt_heart$slp == 0, ]
str(dt_heart_pendiente)
```

```
## 'data.frame':    21 obs. of  14 variables:
##  $ age     : int  63 37 66 53 54 46 59 42 56 62 ...
##  $ sex     : int  1 1 0 1 1 0 1 1 1 0 ...
##  $ cp      : int  3 2 3 2 2 2 3 2 1 0 ...
##  $ trtbps  : int  145 130 150 130 125 142 178 120 120 140 ...
##  $ chol    : int  233 250 226 197 273 177 270 240 240 268 ...
##  $ fbs     : int  1 0 0 1 0 0 0 1 0 0 ...
##  $ restecg : int  0 1 1 0 0 0 0 1 1 0 ...
##  $ thalachh: int  150 187 114 152 152 160 145 194 169 160 ...
##  $ exng    : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 2.6 1.2 0.5 1.4 4.2 0.8 0 3.6 ...
##  $ slp     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ caa     : int  0 0 0 0 1 0 0 0 0 2 ...
##  $ thall   : int  1 2 2 2 2 2 3 3 2 2 ...
##  $ output  : int  1 1 1 1 1 1 1 1 1 0 ...
```

2.4.2 Comprobación de la normalidad y homogeneidad de la varianza.

Para comprobar la normalidad de los distintos atributos que conforman el dataset se analizará una por una, con el fin de aplicar el estudio a aquellas que presenten normalidad. Se aplicarán distintas pruebas de normalidad sobre los datos siendo éstas:

Anderson-Darling: Estudio de bondad de ajuste que mide el área entre la línea ajustada y la función de distribución empírica.

Lilliefors: Es una mejora de la prueba de Kolomogorov-Smirnov y se recomienda usar en conjunto de datos de más de 50 observaciones.

```
# Identificar normalidad en variables del Dataset
library(nortest)
alpha = 0.05
col.names = colnames(dt_heart)
for (i in 1:ncol(dt_heart)) {
  if (i == 1) cat("Variables que no siguen una distribución normal mediante la 'Prueba de Anderson-Darli
  if (is.integer(dt_heart[,i]) | is.numeric(dt_heart[,i])) {
    p_val = ad.test(dt_heart[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if ( (i < ncol(dt_heart) - 1) | (i == ncol(dt_heart)-1) ) cat(", ")
      if (i %% 5 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal mediante la 'Prueba de Anderson-Darling':
## age, sex, cp, trtbps, chol,
## fbs, restecg, thalachh, exng, oldpeak,
## slp, caa, thall, output
```

```
library(nortest)
alpha = 0.05
col.names = colnames(dt_heart)
for (i in 1:ncol(dt_heart)) {
  if (i == 1) cat("Variables que no siguen una distribución normal mediante la 'Prueba de Lilliefors ':
  if (is.integer(dt_heart[,i]) | is.numeric(dt_heart[,i])) {
    p_val = lillie.test(dt_heart[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if ( (i < ncol(dt_heart) - 1) | (i == ncol(dt_heart)-1) ) cat(", ")
      if (i %% 5 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal mediante la 'Prueba de Lilliefors ':
## age, sex, cp, trtbps, chol,
## fbs, restecg, thalachh, exng, oldpeak,
## slp, caa, thall, output
```

Si nos apoyamos tanto en las gráficas de las variables mostradas en un apartado anterior y en la evaluación de los resultados de las pruebas de Anderson-Darling y Lilliefors (para un nivel de significación de 0.05 se ha obtenido p-values inferioriores) podemos afirmar la ausencia de normalidad en los datos de estudio. Definir la ausencia o no, de normalidad es básico para las futuras pruebas estadísticas o modelos de regresión.

A continuación, dado que no se cumple la condición de normalidad en los datos, estudiamos la homogeneidad de varianzas mediante la aplicación de un test de Fligner-Killeen. En este caso, estudiaremos esta homogeneidad en cuanto al grupo conformado por personas con glucemia en ayunas mayor que 120 mg/dl frente a personas con glucemia en ayunas menor que 120mg/dl. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales.

```
fligner.test(output ~ fbs, data = dt_heart)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  output by fbs
## Fligner-Killeen:med chi-squared = 0.23754, df = 1, p-value = 0.626
```

Puesto que obtenemos un p-valor superior a 0'05, aceptamos la hipótesis de que las varianzas de ambas muestras son homogéneas.

2.4.3 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.

¿Qué variables tienen mayor influencia para determinar la presencia de enfermedad cardíaca?

Vamos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre la posible presencia de ataque cardíaco. En este caso, dado que los datos no siguen

una distribución normal, utilizaremos el coeficiente de correlación de Spearman.

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")

# Calcular el coeficiente de correlación para cada variable
# con respecto al campo "output"
for (i in 1:(ncol(dt_heart) - 1)) {

    spearman_test = cor.test(dt_heart[,i],
                             dt_heart[,length(dt_heart)],
                             method = "spearman", exact = FALSE)
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Add row to matrix
    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(dt_heart)[i]
}
print(corr_matrix)
```

```
##             estimate      p-value
## age      -0.23840007 2.749629e-05
## sex      -0.28093658 6.678692e-07
## cp        0.46086018 2.444718e-17
## trtbps   -0.12159275 3.437373e-02
## chol     -0.12088824 3.543860e-02
## fbs      -0.02804576 6.267775e-01
## restecg   0.14861154 9.581603e-03
## thalachh  0.42836989 5.938298e-15
## exng     -0.43675708 1.520814e-15
## oldpeak  -0.42148706 1.766192e-14
## slp       0.37146048 2.393250e-11
## caa      -0.49935521 1.631269e-20
## thall    -0.41484737 4.938190e-14
```

Así, identificamos cuáles son las variables más correlacionadas con un diagnóstico de enfermedad cardíaca en función de su proximidad con los valores -1 y +1. Teniendo esto en cuenta, queda patente cómo la variable más relevante en el diagnóstico de enfermedad cardíaca es el número de vasos sanguíneos principales coloreados por fluoroscopia (caa), que está correlacionada negativamente con la variable objetivo, seguida del tipo de dolor en el pecho (cp) correlacionada positivamente, la presencia de angina inducida por el ejercicio (exng) correlacionada negativamente, la frecuencia cardíaca máxima alcanzada (thalachh) correlacionada positivamente, y en último lugar la depresión del ST inducida por el ejercicio en relación con el reposo (oldpeak), correlacionada negativamente con la variable objetivo.

¿Es propenso el paciente a tener una enfermedad cardíaca si la glucemia en ayunas es > 120 mg/dl?

La segunda prueba estadística que se aplicará consistirá en un contraste de hipótesis para comprobar si existen diferencias significativas en la variable objetivo (output) entre los grupos definidos por la variable Glucemia en ayunas > 120 mg/dl (fbs), con el fin de averiguar si el paciente es más o menos propenso a la presencia de enfermedad cardíaca según su nivel de glucemia en ayunas. Como ambas variables son categóricas y los datos no cumplen la condición de normalidad, lo más eficiente es aplicar el test Chi-cuadrado de Pearson para comprobar si existen diferencias significativas entre ambas variables. Así, se plantea el siguiente contraste de hipótesis sobre la independencia de la propensión a tener un ataque cardíaco sobre el índice de glucemia en

ayunas:

H0 : La probabilidad de sufrir un ataque cardíaco es independiente de los niveles de glucemia en ayunas

H1 : La probabilidad de sufrir un ataque cardíaco no es independiente de los niveles de glucemia en ayunas

```
# Tabla de contingencia
heart_output_fbs <- table(dt_heart$output, dt_heart$fbs)
# Test Chi-cuadrado de independencia entre output y fbs
chisq.test(heart_output_fbs, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  heart_output_fbs
## X-squared = 0.23833, df = 1, p-value = 0.6254
```

A partir de las frecuencias de cada valor del estado de enfermedad angiográfico para cada uno de los grupos, se observa que obtenemos un p-valor mayor que el valor de significación (0,05), por tanto no se rechaza la hipótesis nula. Así, no tenemos suficientes pruebas para concluir que si la glucemia del paciente en ayunas es > 120 mg/dl, el paciente sea propenso a poseer una enfermedad cardíaca.

¿Es propenso el paciente a tener una enfermedad cardíaca si la pendiente del segmento ST de ejercicio máximo es hacia arriba?

Esta prueba consistirá en un contraste de hipótesis para comprobar si existen diferencias significativas en la variable objetivo entre los grupos definidos por la variable Pendiente del segmento ST de ejercicio máximo (slp), con el fin de averiguar si el paciente es más o menos propenso a la presencia de enfermedad cardíaca según el valor de la pendiente del segmento ST. Nos encontramos en las mismas condiciones que el caso anterior, así que plantearemos un contraste de hipótesis sobre la independencia de la propensión a tener un ataque cardíaco sobre la pendiente del segmento ST:

H0 : La probabilidad de sufrir un ataque cardíaco es independiente del valor de la pendiente del segmento ST.

H1 : La probabilidad de sufrir un ataque cardíaco no es independiente del valor de la pendiente del segmento ST.

```
# Test Chi-cuadrado de independencia entre output y slp
heart_output_slp <- table(dt_heart$output, dt_heart$slp)
chisq.test(heart_output_slp, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  heart_output_slp
## X-squared = 47.507, df = 2, p-value = 4.831e-11
```

En este caso, se observa que obtenemos un p-valor mucho menor que el valor de significación fijado, así que rechazamos la hipótesis nula. Por tanto, podemos concluir que, efectivamente, el paciente es más propenso a padecer un ataque cardíaco si la pendiente del segmento ST es hacia arriba. ## Modelo de regresión logística Tal y como se planteó en los objetivos de la actividad, resultará de mucho interés poder realizar predicciones sobre las probabilidades del paciente a sufrir un ataque cardíaco dadas sus características. Así, se calculará un modelo de regresión logística utilizando regresores tanto cuantitativos como cualitativos con el que poder realizar las predicciones de las probabilidades. Para obtener un modelo de regresión logística considerablemente eficiente, lo que haremos será obtener varios modelos de regresión utilizando las variables que estén más correlacionadas con la variable de salida, según los resultados obtenidos anteriormente. Así, de entre todos los modelos que tengamos, escogeremos el mejor utilizando como criterio aquel que presente un menor AIC.

```r
# Regresores cuantitativos con mayor coeficiente de correlación con respecto a la probabilidad de ataqu
#caa thalachh oldpeak
#Regresores cualitativos
#cp exng thall slp
# Variable a predecir
#$output

#1-fold Train-Test Split
set.seed(4321)
indexes <- sample(1:nrow(dt_heart), floor(.7*nrow(dt_heart)))

dt_heart.train <- dt_heart[indexes,]
dt_heart.test <- dt_heart[-indexes,]

# Generación de varios modelos
modelo1 <- glm(formula = output ~ caa + thalachh + oldpeak + cp + exng, data = dt_heart.train, family=bi
modelo2 <- glm(formula = output ~ caa + thalachh + oldpeak + cp + exng + thall + slp, data = dt_heart.tr
modelo3 <- glm(formula = output ~ caa + cp + exng, data = dt_heart.train, family=binomial(link="logit"))
modelo4 <- glm(formula = output ~ caa + thalachh + oldpeak, data = dt_heart.train,  family=binomial(link
```

En este caso, la bondad del modelo se evaluará mediante la medida AIC. Dado que esta medida tiene en cuenta tanto la bondad del ajuste (el error) como la complejidad del modelo, seleccionaremos aquel modelo que resulte en el menor AIC.

```r
#  AIC de cada modelo
summary(modelo1)$aic
```

```
## [1] 192.4814
```

```r
summary(modelo2)$aic
```

```
## [1] 183.0943
```

```r
summary(modelo3)$aic
```

```
## [1] 212.2853
```

```r
summary(modelo4)$aic
```

```
## [1] 212.3726
```

Observamos que el mejor modelo es el segundo, ya que es el que posee menor AIC:

```r
summary(modelo2)
```

```
##
## Call:
## glm(formula = output ~ caa + thalachh + oldpeak + cp + exng +
##     thall + slp, family = binomial(link = "logit"), data = dt_heart.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5359  -0.4819   0.2649   0.6256   2.4316
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.755579   1.654467   0.457 0.647893
## caa         -1.017496   0.250352  -4.064 4.82e-05 ***
```

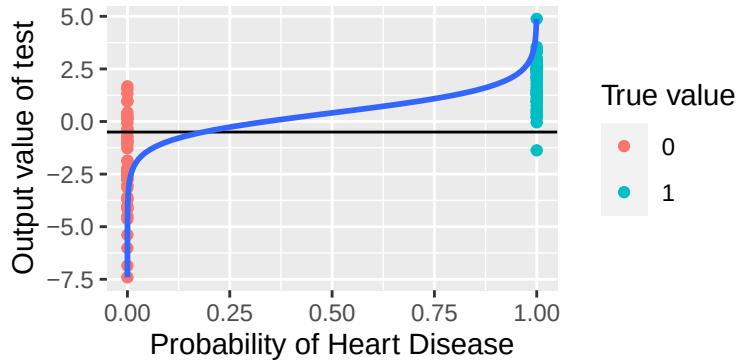
```
## thalachh     0.016295   0.009894   1.647 0.099565 .
## oldpeak     -0.654002   0.251203  -2.603 0.009228 **
## cp           0.558657   0.207376   2.694 0.007062 **
## exng        -0.873063   0.450074  -1.940 0.052401 .
## thall       -1.139052   0.338120  -3.369 0.000755 ***
## slp          0.492043   0.387051   1.271 0.203635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 291.17  on 211  degrees of freedom
## Residual deviance: 167.09  on 204  degrees of freedom
## AIC: 183.09
##
## Number of Fisher Scoring iterations: 5
```

Con estos valores dibujamos la gráfica del modelo de regresión logística:

```
library(magrittr, warn.conflicts = FALSE)
library(dplyr, warn.conflicts = FALSE)

dt_heart_2 <- dt_heart[-indexes,] %>% mutate(test = 0.755579 -1.017496*caa
                                + 0.016295 * thalachh
                                - 0.654002 * oldpeak
                                + 0.558657 * cp
                          - 0.873063 * exng
                          - 1.139052 * thall
                          + 0.492043 * slp)
ggplot(dt_heart_2, aes(y = output, x= test)) +
            geom_point(mapping = aes(col=as.factor(output))) +
            geom_vline(xintercept=-0.5) +
            geom_smooth(method = "glm",
                                method.args = list(family = "binomial"),
                                se = FALSE) +
            coord_flip() +
            labs( x = "Output value of test",
                  y = "Probability of Heart Disease",
                  col = "True value")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Ahora, utilizando este modelo, podemos realizar predicciones sobre la probabilidad de sufrir un ataque cardíaco. Realizamos predicciones sobre los

conjuntos de entrenamiento y test, con el objetivo de calcular la precisión en cada uno:

```
#Importar libreria requerida
library(caret, warn.conflicts = FALSE)
```

```
## Loading required package: lattice
```

```
# Prediccion sobre los conjuntos de train y test
dt_heart.train.predict<-predict(modelo2,dt_heart.train)
dt_heart.test.predict<-predict(modelo2,dt_heart.test)
#Train accuracy for model:
mean(as.numeric(dt_heart.train.predict>=0)==dt_heart.train$output)
```

```
## [1] 0.8584906
```

```
#Test accuracy for model
mean(as.numeric(dt_heart.test.predict>=0)==dt_heart.test$output)
```

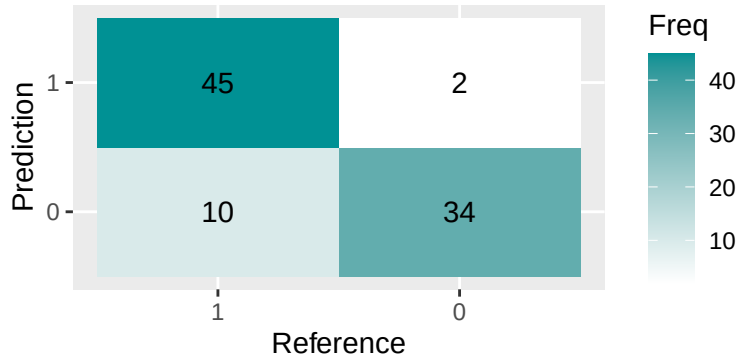
```
## [1] 0.8681319
```

Una vez calculados los valores predichos, pintamos la matriz de confusión:

```
# Matriz de confusion
cm <- confusionMatrix(data=factor(dt_heart.test$output), reference=factor(as.numeric(dt_heart.test.predi

plt <- as.data.frame(cm$table)
plt$Prediction <- factor(plt$Prediction, levels=rev(levels(plt$Prediction)))

ggplot(plt, aes(Prediction,Reference, fill= Freq)) +
        geom_tile() + geom_text(aes(label=Freq)) +
        scale_fill_gradient(low="white", high="#009194") +
        labs(x = "Reference",y = "Prediction") +
        scale_x_discrete(labels=c("1","0")) +
        scale_y_discrete(labels=c("0","1"))
```



Observando la matriz de confusión, vemos que el modelo predice correctamente 79 de los 91 diagnósticos de pacientes, de los cuales 45 son clasificados correctamente como propensos a sufrir un ataque al corazón y 34 poseen una situación no crítica. Además, 10 pacientes son clasificados incorrectamente como poco propensos a enfermedad cardíaca y 2 pacientes son clasificados como propensos a la enfermedad, cuando en realidad no lo son.

Observamos también las medidas de bondad del modelo:

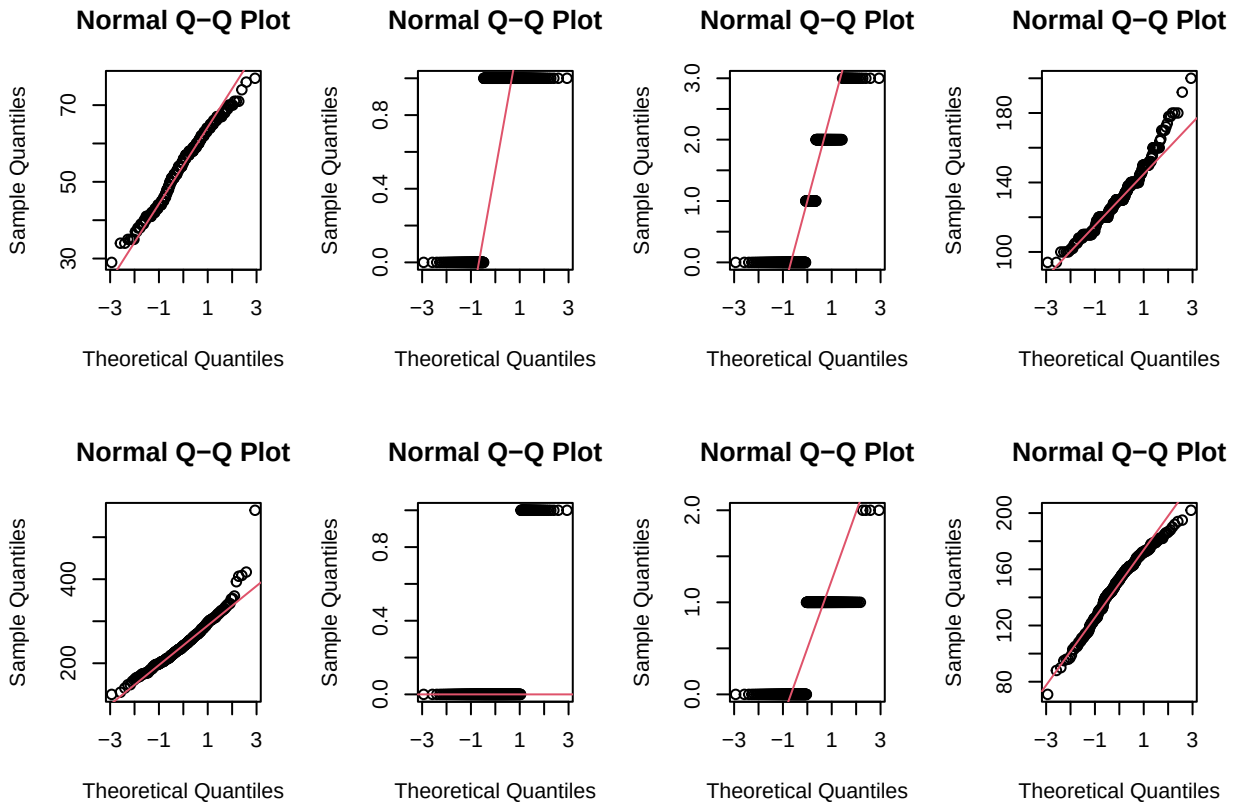
```
confusionMatrix(data=factor(dt_heart.test$output), reference=factor(as.numeric(dt_heart.test.predict>=0
```

```
## Confusion Matrix and Statistics
##
##          Prediction
```

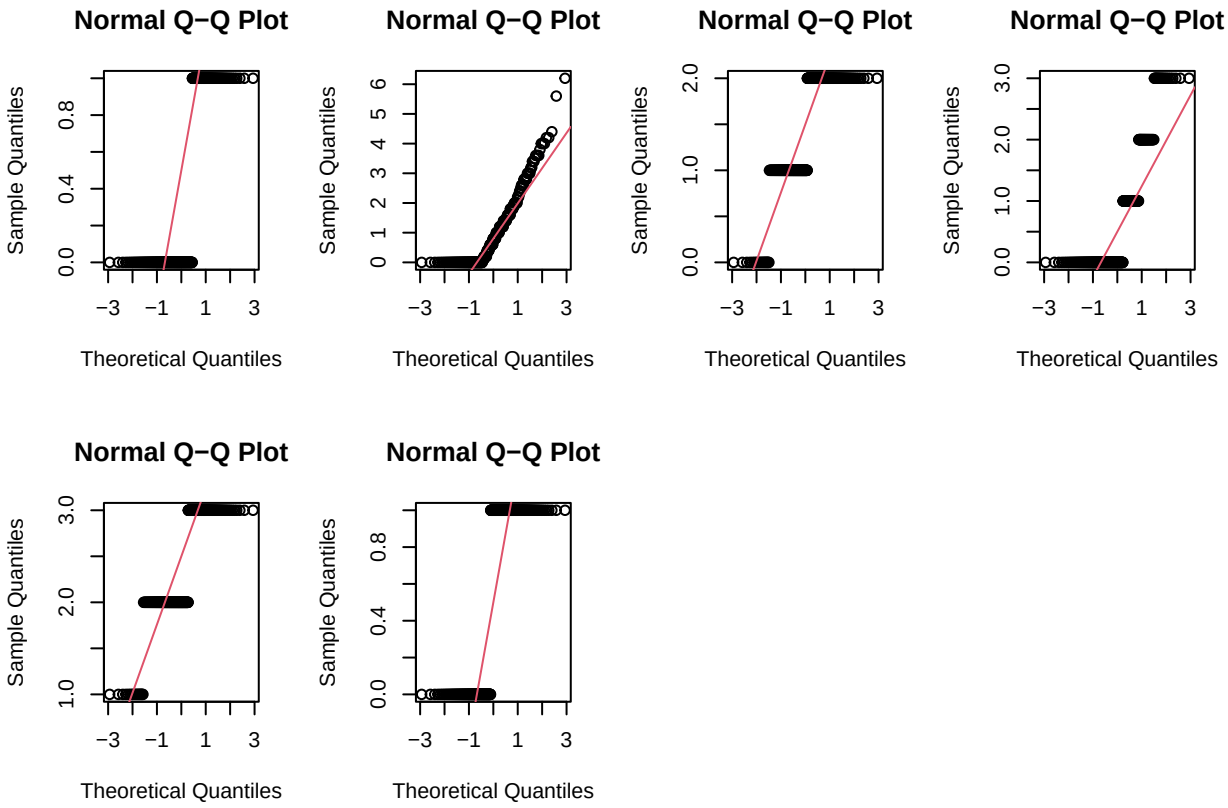
```
## Reference  0  1
##         0 34 10
##         1  2 45
##
##                Accuracy : 0.8681
##                  95% CI : (0.781, 0.93)
##     No Information Rate : 0.6044
##     P-Value [Acc > NIR] : 3.164e-08
##
##                   Kappa : 0.7344
##
##  Mcnemar's Test P-Value : 0.04331
##
##             Sensitivity : 0.9444
##             Specificity : 0.8182
##          Pos Pred Value : 0.7727
##          Neg Pred Value : 0.9574
##              Prevalence : 0.3956
##          Detection Rate : 0.3736
##    Detection Prevalence : 0.4835
##       Balanced Accuracy : 0.8813
##
##        'Positive' Class : 0
##
```

En este caso, podríamos decir que el modelo funciona bastante bien, puesto que tanto la sensibilidad como la especificidad son altas. Una prueba altamente sensible significa que hay pocos resultados falsos negativos y, por lo tanto, se pasan por alto menos casos propensos a la enfermedad. La especificidad de una prueba es su capacidad para designar como negativo a un individuo que no es propenso a una enfermedad. Una prueba altamente específica significa que hay pocos resultados falsos positivos. ## Representación de los resultados a partir de tablas y gráficas. Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica. A lo largo de la práctica, se ha realizado una exploración visual de las distintas variables y representaciones para analizar los valores extremos o vacíos. En el apartado 4 hemos visto que nuestros datos no cumplen la condición de normalidad. Para representar la normalidad de los datos se pueden utilizar los QQplots ya que permiten observar la similitud entre las distribuciones de dos conjuntos de datos, la analizada y una distribución normal ideal. Así, realizamos una representación de cada variable del dataset:

```
par(mfrow=c(2,4))
qqnorm(dt_heart$age)
qqline(dt_heart$age,col=2)
qqnorm(dt_heart$sex)
qqline(dt_heart$sex,col=2)
qqnorm(dt_heart$cp)
qqline(dt_heart$cp,col=2)
qqnorm(dt_heart$trtbps)
qqline(dt_heart$trtbps,col=2)
qqnorm(dt_heart$chol)
qqline(dt_heart$chol,col=2)
qqnorm(dt_heart$fbs)
qqline(dt_heart$fbs,col=2)
qqnorm(dt_heart$restecg)
qqline(dt_heart$restecg,col=2)
qqnorm(dt_heart$thalachh)
qqline(dt_heart$thalachh,col=2)
```




```
qqnorm(dt_heart$exng)
qqline(dt_heart$exng,col=2)
qqnorm(dt_heart$oldpeak)
qqline(dt_heart$oldpeak,col=2)
qqnorm(dt_heart$slp)
qqline(dt_heart$slp,col=2)
qqnorm(dt_heart$caa)
qqline(dt_heart$caa,col=2)
qqnorm(dt_heart$thall)
qqline(dt_heart$thall,col=2)
qqnorm(dt_heart$output)
qqline(dt_heart$output,col=2)
```



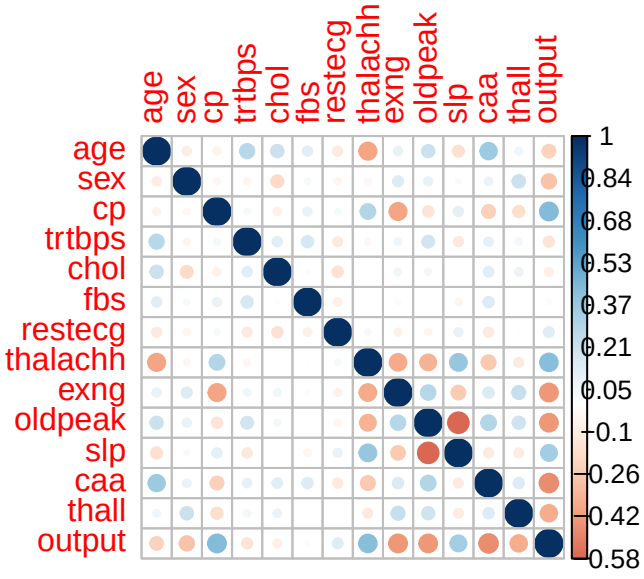
Como podemos observar, todas las distribuciones se alejan mucho de la normal.

Por otra parte, representaremos una matriz de correlación para visualizar la correlación entre las variables:

```
# Cargamos librería
library(corrplot)
```

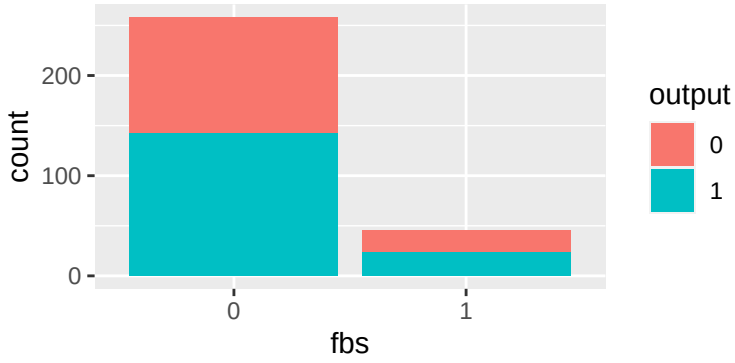
```
## corrplot 0.92 loaded
```

```
# Pintamos la el gráfico
corrplot(cor(dt_heart), is.cor = FALSE)
```



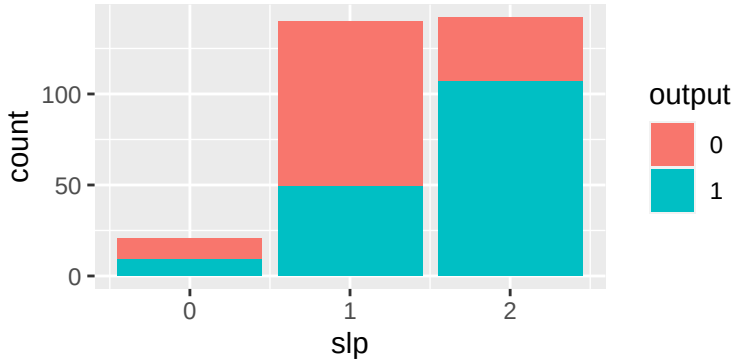
En este gráfico podemos observar como efectivamente, las variables más correlacionadas con nuestra variable objetivo son: caa, thalach, oldpeak, cp, exng, thall y slp, ya que son las que tienen un mayor y más marcado círculo. En cuanto a los contrastes de hipótesis, las diferencias encontradas entre los grupos se pueden analizar a través de diagramas de barras, para ver más claramente la dirección de estas diferencias. Así, en este caso se puede observar que no se aprecian grandes diferencias cuando la glucemia en ayunas es mayor que 120 mg/dl, ya que la frecuencia es similar para propensos a ataque cardíaco y para situaciones no críticas.

```
# Convertir variables a factor
dt_heart$fbs <- as.factor(dt_heart$fbs)
dt_heart$output <- as.factor(dt_heart$output)
# Reprentacion gráfica
ggplot(data = dt_heart, aes(x = fbs, fill = output)) + geom_bar()
```



Por otro lado,se aprecia una mayor frecuencia a que el diagnóstico sea enfermedad cardíaca si la pendiente del segmento ST de ejercicio máximo es hacia arriba, como podemos observar en el diagrama:

```
# Convertir variables a factor
dt_heart$fbs <- as.factor(dt_heart$slp)
dt_heart$output <- as.factor(dt_heart$output)
# Reprentacion gráfica
ggplot(data = dt_heart, aes(x = slp, fill = output)) + geom_bar()
```



En el caso de la regresión/clasificación logística, nuevamente se ha optado por realizar la representación del modelo de regresión logística y la matriz de confusión en el propio apartado.

```
# Exportar el fichero resultante con los datos finales analizados
write.csv(dt_heart, "heart_final.csv", row.names=FALSE)
```

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

En primer lugar, se ha realizado una exploración visual para comprobar la distribución de los datos y se han tratado valores nulos en las variables. Los valores outliers no se han tratado como tal, ya que son valores reales y su supresión llevaría a una interpretación errónea. De los métodos de contraste de pótesis y correlaciones, se puede observar que los pacientes con la pendiente del segmento ST de ejercicio máximo hacia arriba son más propensos a padecer un ataque cardíaco. Además, se ha verificado que la glucemia en ayunas > 120 mg/dl no es un factor significativo a la hora de diagnosticar la propensión a esta enfermedad. Se ha confirmado que la variable más significativa a la hora de detectar un posible ataque cardíaco es el número de vasos sanguíneos coloreados por fluoroscopia (caa), ya que en este caso se puede detectar una posible obstrucción en algún tejido que no sea coloreado. Algunas de las variables cuantitativas cuyos valores pueden ser indicativo de aumentar/reducir la probabilidad a padecer la enfermedad son la frecuencia cardíaca máxima alacanzada (thalachh), la depresión del ST inducida por el ejercicio en relación con el reposo (oldpeak). En cuanto a las variables categóricas, cobran importancia el tipo de dolor en el pecho (cp), la presencia de angian inducida por el ejercicio (exng), el resultado de la prueba de esfuerzo con talio (thall) y la pendiente del segmento ST de ejercicio máximo (slp). Posteriormente, con estas variables se han entrenado varios modelos de regresión logística, eligiendo finalmente el de mayor bondad, y realizando predicciones sobre el conjunto de test. El modelo funciona bastante bien, puesto que tanto la sensibilidad como la especificidad son altas, y posee una precisión de 0.8681. Es decir, la probabilidad de que el modelo clasifique correctamente a una persona propensa a padecer un ataque cardíaco es de +-87%.

```
library(knitr)
df <- data.frame(Contribuciones = c("Investigación previa","Redacción de las respuestas","Desarrollo del
                 Firma = c("Alba Caderno Fernández, Diego García García","Alba Caderno Fernández, Diego
kable(df)
```

Contribuciones	Firma
Investigación previa	Alba Caderno Fernández, Diego García García
Redacción de las respuestas	Alba Caderno Fernández, Diego García García
Desarrollo del código	Alba Caderno Fernández, Diego García García
Participación en el vídeo	Alba Caderno Fernández, Diego García García