Práctica 2 – Limpieza y análisis de datos

Adrián González González

HTTPS://GITHUB.COM/ADRIANGLEZ77/LIMPIEZA-Y-AN-LISIS-DE-DATOS.GIT

1.- Descripción del dataset.

El dataset escogido para el desarrollo de la práctica pertenece al ofrecido en el enunciado de ella. Se denomina "Heart Attack Analysis & Prediction", y se encuentra disponible en el repositorio Kaggle:

Heart Attack Analysis & Prediction Dataset | Kaggle

El dataset tiene 14 atributos que recogen información personal sobre pacientes, y los resultados que dichos pacientes han obtenido en diversas pruebas sanitarias. Esta información permite resolver el problema de la detección temprana de enfermedades del corazón utilizando los atributos del dataset que permitan realizar la predicción.

Los atributos que conforman el conjunto de datos son los siguientes:

- age: edad del paciente.
- sex: sexo del paciente (1 o 0).
- cp: tipo de dolor en el pecho:
    - 1: angina típica
    - 2: angina atípica
    - 3: no dolor angina
    - 4: asintomático
- trtbps: presión arterial en reposo (medido en mm Hg).
- chol: colesterol en mg/dl obtenido a través del sensor de IMC.
- fbs: azúcar en sangre en ayunas > 120 mg/dl
    - 1 = verdadero
    - 0 = falso
- restecg: resultados electrocardiográficos en reposo:
    - 0 = normal
    - 1 = tener anomalía de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST de > 0,05 mV)
    - 2 = hipertrofia ventricular izquierda probable o definida según los criterios de Estes

- thalach: ritmo cardiaco máximo alcanzado.
- exng: angina inducida por el ejercicio:
    - 1 = sí
    - 0 = no
- oldpeak: Depresión ST inducida por el ejercicio en relación con el descanso.
- slp: pendiente:
    - 0 = sin pendiente
    - 1 = plano
    - 2 = pendiente descendente
- caa: número de vasos principales (0-3).
- thall: tasa de talasemia:
    - 0 = nulo
    - 1 = defecto corregido
    - 2 = normal
    - 3 = defecto reversible
- output: variable predictora:
    - 0: < 50% de estrechamiento del diámetro. menos probabilidad de enfermedad cardíaca
    - 1: > 50% de estrechamiento del diámetro. más probabilidad de enfermedad cardíaca

Para entender un poco más algunos conceptos, vamos a definirlos:

- Angina: dolor en el pecho debido a la reducción del flujo sanguíneo a los músculos del corazón.
- Colesterol: sustancia cerosa que se encuentra en las células del cuerpo y pertenece a un grupo de moléculas orgánicas llamadas lípidos. Hay 3 tipos de colesterol; lipoproteína de alta densidad (HDL) y se conoce como el "colesterol bueno", lipoproteína de baja densidad (LDL) conocida como el "colesterol malo" y lipoproteínas de muy baja densidad (VLDL) y como su nombre lo indica, son partículas de baja densidad que transportan triglicéridos en la sangre.
- ECG: abreviatura de electrocardiograma, es una prueba de rutina que generalmente se realiza para verificar la actividad eléctrica del corazón.
- Talasemia: es un trastorno genético de la sangre que se caracteriza por una tasa de hemoglobina más baja de lo normal.

2.- Integración y selección

La parte de integración está resulta en el código en el lenguaje “R”, donde se ha pasado desde la primera etapa de incluir las librerías, cargar el dataset, visualizar las variables y el tipo de datos que contienen, estructura del conjunto de datos, y por último categorizar algunas variables.

Se cargan algunas librerías básicas para generar gráficos y para manejar los datos.

```
library(ggplot2)
library(dplyr)
```

Para comenzar, cargaremos el dataset y visualizaremos las primeras filas.

```
heart <- read.csv("H:/Mi unidad/Tipología y ciclo de vida de los datos/PRA 2 - 13ene
/heart.csv")
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
```

Con la función “summary”, vamos a obtener un breve resumen estadístico de las variables.

```
summary(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
```

El siguiente paso será ver la estructura del conjunto de datos, con la función “str”.

```
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 ...
```

Se observa que hay 303 registros y 14 variables. Todas las variables se han clasificado automáticamente como variables "integer" y una como "numeric", por lo cual tendremos que transformar y categorizar las variables discretas a tipo "factor". Estas variables son: "sex", "cp", "fbs", "rest_ecg", "exang", "slp", "caa", "thall", y "output".

La primera variable con la trabajaremos será "sex", ya que el 0 indica a una mujer y el 1 a un hombre, lo pondremos como factor.

```
heart$sex <- as.factor(heart$sex)

heart$sex <- factor(heart$sex, levels=c(0,1), labels=c("mujer", "hombre"))

head(heart)
```

```
##    age      sex cp trtbps chol fbs restecg thalachh exng oldpeak slp caa thall
## 1   63 hombre   3    145   233   1       0      150    0     2.3   0   0     1
## 2   37 hombre   2    130   250   0       1      187    0     3.5   0   0     2
## 3   41  mujer   1    130   204   0       0      172    0     1.4   2   0     2
## 4   56 hombre   1    120   236   0       1      178    0     0.8   2   0     2
```

Lo siguiente será factorizar las demás variables discretas.

```
heart$cp <- as.factor(heart$cp)
heart$fbs <- as.factor(heart$fbs)
heart$restecg <- as.factor(heart$restecg)
heart$exng <- as.factor(heart$exng)
heart$slp <- as.factor(heart$slp)
heart$caa <- as.factor(heart$caa)
heart$thall <- as.factor(heart$thall)
heart$output <- as.factor(heart$output)
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 "mujer","hombre": 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     : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ 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 ...
```

La gran mayoría de los atributos presentes en el conjunto de datos se corresponden con características que reúnen los pacientes recogidos en forma de registros, por lo que será conveniente tenerlos en consideración durante la realización de los análisis. En este principio, no prescindiremos de ningún atributo hasta tener una visión más detallada de la influencia de cada uno de ellos en las enfermedades del corazón.

3.- Limpieza de los datos.

3.1- ¿Los datos contienen ceros o elementos vacíos? Gestiona cada uno de estos casos

Para esta parte de limpieza de datos, lo más habitual es eliminar los registros que poseen "0" para indicar la ausencia de ciertos valores, pero en nuestro caso, al tratar con variables categóricas en la que estos "0" tienen importancia y un significado, no se eliminarán ni se buscarán.

Por ello, nos centraremos en conocer los campos que contienen elementos nulos (NA, del inglés, Not Available) y en las cadenas de texto que tienen elementos vacíos.

```
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
```

```
colSums(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
```

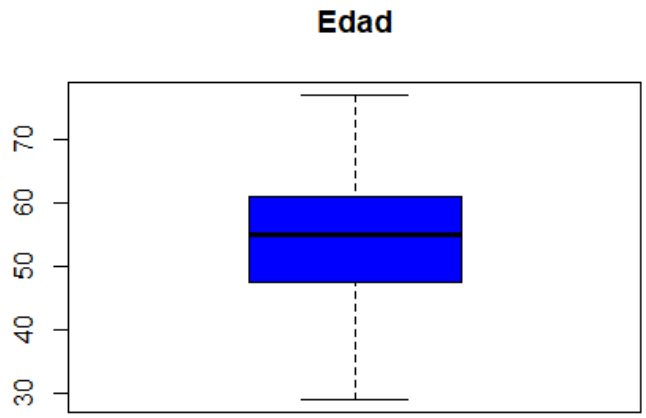
Como se aprecia en el resultado de la práctica, no tenemos valores nulos ni valores vacíos. En caso de haberlos, tendríamos que decidir cómo manejar estos campos (eliminar los registros, asignar valores, etc).

3.2- Identifica y gestiona los valores extremos.

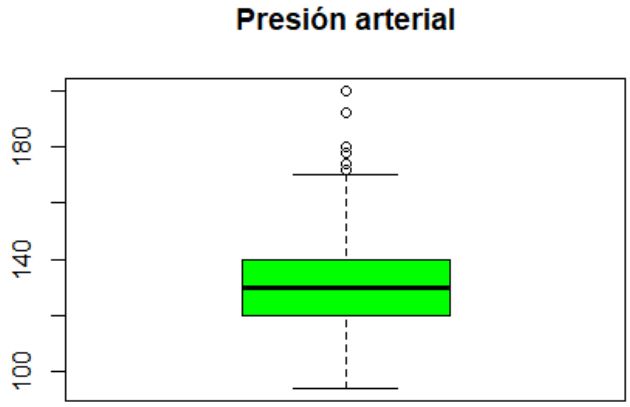
Los valores extremos o outliers son aquellos que parecen no ser congruentes sin los comparamos con el resto de los datos. Estos outliers son observaciones que se desvían mucho de otras y a veces, pueden suponer un problema tenerlos en el conjunto de datos. Es por ello, que hay que realizar un análisis, primero para identificar si los hay, y segundo para identificar la posible causa y ver si tienen que ser eliminados o pueden permanecer en el dataset.

Para identificarlos, vamos a representar cada variable continua en un diagrama de caja y ver qué valores distan mucho del rango intercuartílico (la caja).

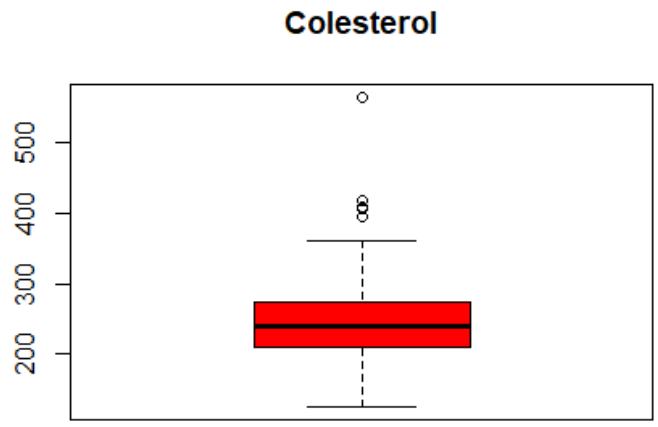
```r
boxplot(heart$age,main = "Edad",col="blue")
```



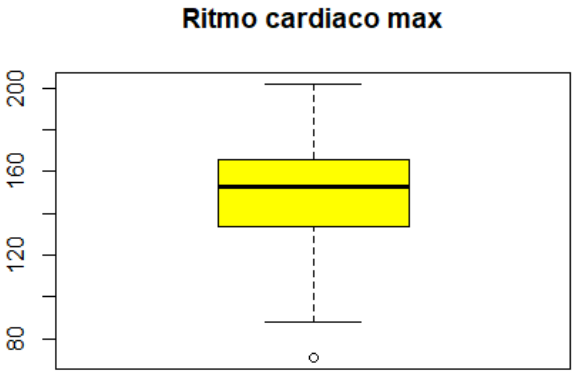

```r
boxplot(heart$trtbps,main = "Presión arterial",col="green")
```



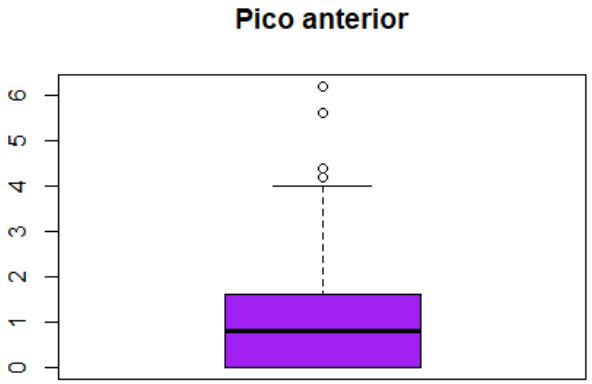

```r
boxplot(heart$chol,main = "Colesterol",col="red")
```




```
boxplot(heart$thalachh,main = "Ritmo cardiaco max",col="yellow")
```




```
boxplot(heart$oldpeak,main = "Pico anterior",col="purple")
```



Tras observar los resultados gráficamente de las variables “age”, “trtbps”, “chol”, “thalchh” y “oldpeak”, el único atributo sin valores extremos es la que referencia a la edad.

Todas las demás variables poseen algunos valores fuera de lo “normal”, aunque la presión arterial “trtbps”, y colesterol “chol” tienen unos pocos valores por encima del rango intercuartílico. La variable que hace referencia al ritmo cardiaco máximo “thalachh”, tiene un valor bastante por debajo de dicha caja.

No se van a eliminar estos datos del conjunto de datos, ya que, al estar estudiando la influencia de estas variables en la posibilidad de sufrir una enfermedad de corazón, es de especial importancia saber si los pacientes que se salen de este rango en una o varias variables, tienen más riesgo de sufrirla.

Pero como estudio, vamos a observar cuáles son estos “outliers” de las variables que hemos mencionado anteriormente.

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

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

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

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

```
boxplot.stats(heart$thalachh)$out
```

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

Para finalizar con esta etapa, se va a exportar a un nuevo fichero “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?)

```
heart["edad_r"] <- cut(heart$age, breaks = c(0,35,45,55,60,65,100),labels = c("<=35"
, "36-45","46-55","56-60","61-65",">65"))

table(heart$edad_r)
```

```
##
##  <=35 36-45 46-55 56-60 61-65   >65
##     7    57    88    72    46    33
```

```
summary(heart)
```

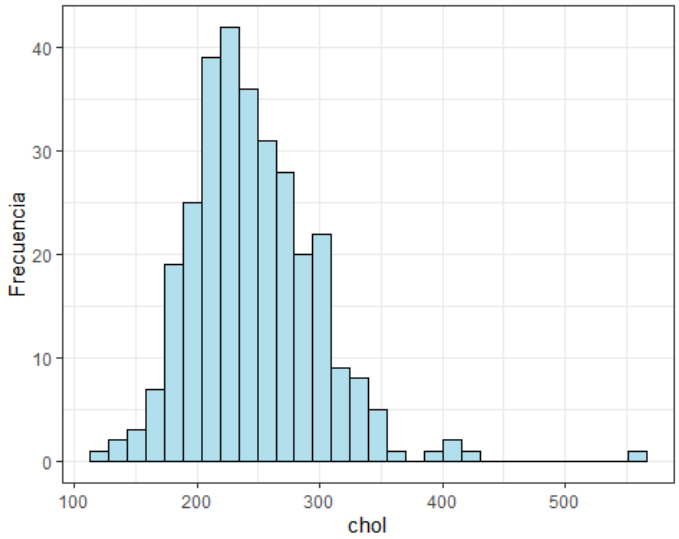
```
##       age            sex      cp          trtbps           chol       fbs
##  Min.   :29.00   mujer : 96   0:143   Min.   : 94.0   Min.   :126.0   0:258
##  1st Qu.:47.50   hombre:207   1: 50   1st Qu.:120.0   1st Qu.:211.0   1: 45
##  Median :55.00                2: 87   Median :130.0   Median :240.0
##  Mean   :54.37                3: 23   Mean   :131.6   Mean   :246.3
##  3rd Qu.:61.00                        3rd Qu.:140.0   3rd Qu.:274.5
##  Max.   :77.00                        Max.   :200.0   Max.   :564.0
##  restecg    thalachh     exng       oldpeak     slp     caa     thall   output
##  0:147   Min.   : 71.0   0:204   Min.   :0.00   0: 21   0:175   0:  2   0:138
```

```
##  1:152   1st Qu.:133.5   1: 99   1st Qu.:0.00   1:140   1: 65   1: 18   1:165
##  2:  4   Median :153.0           Median :0.80   2:142   2: 38   2:166
##          Mean   :149.6           Mean   :1.04           3: 20   3:117
##          3rd Qu.:166.0           3rd Qu.:1.60           4:  5
##          Max.   :202.0           Max.   :6.20
##    edad_r
##  <=35 : 7
##  36-45:57
##  46-55:88
##  56-60:72
##  61-65:46
##  >65  :33
```

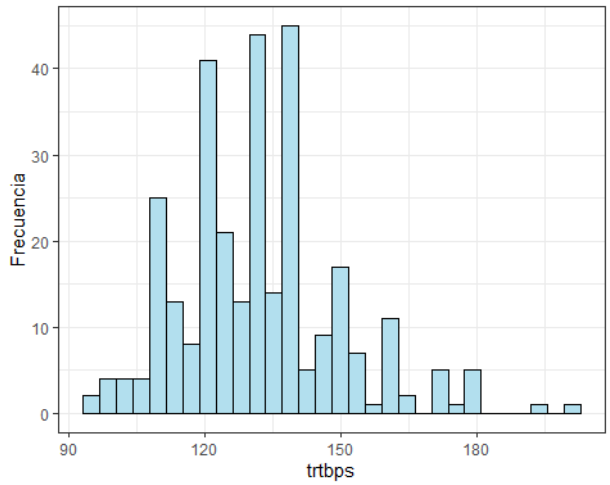
4.2- Comprobación de la normalidad y homogeneidad de la varianza.

Se va a ver gráficamente los valores y distribución de las variables continuas “chol”, “trtbps”, “thalalchh”, “oldpeak”.

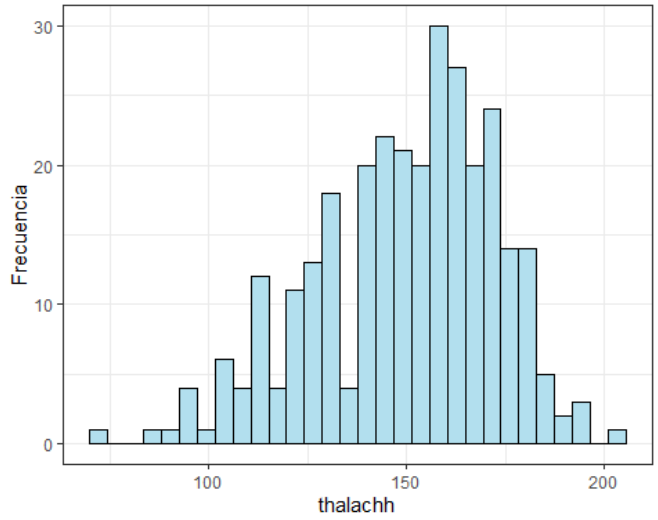
```
ggplot(heart, aes(x=chol)) +
  geom_histogram(fill="lightblue2", colour="black") +
  labs(x="chol", y="Frecuencia") +
  theme_bw()

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



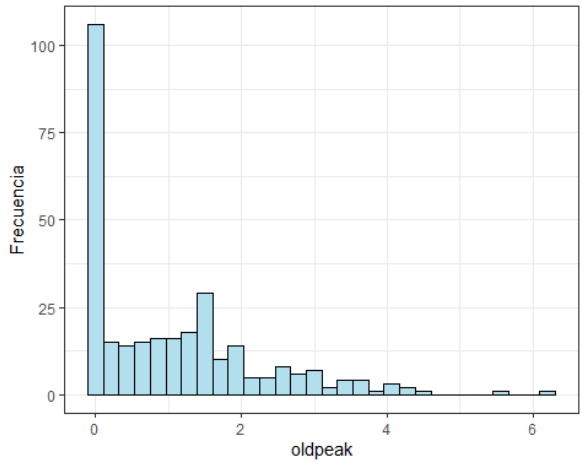

```
ggplot(heart, aes(x=trtbps)) +
  geom_histogram(fill="lightblue2", colour="black") +
    labs(x="trtbps", y="Frecuencia") +
  theme_bw()

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




```
ggplot(heart, aes(x=thalachh)) +
  geom_histogram(fill="lightblue2", colour="black") +
    labs(x="thalachh", y="Frecuencia") +
  theme_bw()

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




```
ggplot(heart, aes(x=oldpeak)) +
  geom_histogram(fill="lightblue2", colour="black") +
    labs(x="oldpeak", y="Frecuencia") +
  theme_bw()

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



Lo que se ve a simple vista en los histogramas, éstos reflejan que nuestros valores no tienen una distribución normal, excepto la variable que caracteriza al colesterol, que sigue cierta normalidad.

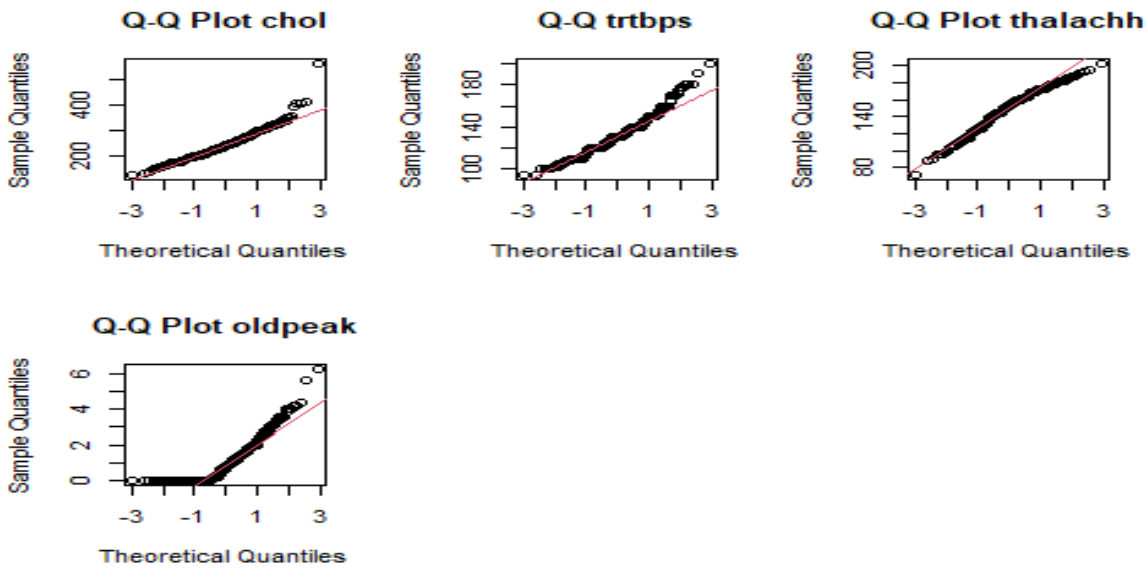
Para comprobar estadísticamente esta normalidad, vamos a hacer uso de algunos test:

Test del gráfico Q-Q.

Muchas pruebas estadísticas suponen que un conjunto de datos sigue una distribución normal y, a menudo, se utiliza una gráfica QQ para evaluar si se cumple o no este supuesto. Aunque un gráfico QQ no es una prueba estadística formal, proporciona una manera fácil de verificar visualmente si un conjunto de datos sigue una distribución normal y, de no ser así, cómo se viola esta suposición y qué puntos de datos pueden causar esta violación.

En nuestro caso, hemos puesto el mismo conjunto de datos tanto para el "qqnorm" como para "qqline", para ver si los datos siguen una línea diagonal recta y siguen la distribución normal, o si se alejan de la línea y no la siguen.

```
library(moments)

## Warning: package 'moments' was built under R version 4.1.3

juntar <- par(mfrow=c(2,3))

qq1 <- qqnorm(heart$chol, main = "Q-Q Plot chol");qqline(
heart$chol, col = 2)

qq2 <- qqnorm(heart$trtbps, main = "Q-Q trtbps");qqline(
heart$trtbps, col = 2)


qq3 <- qqnorm(heart$thalachh, main = "Q-Q Plot thalachh");qqline(
heart$thalachh, col = 2)


qq4 <- qqnorm(heart$old, main = "Q-Q Plot oldpeak");qqline(
heart$old, col = 2)


par(juntar)
```



Test de Shapiro-Wilk

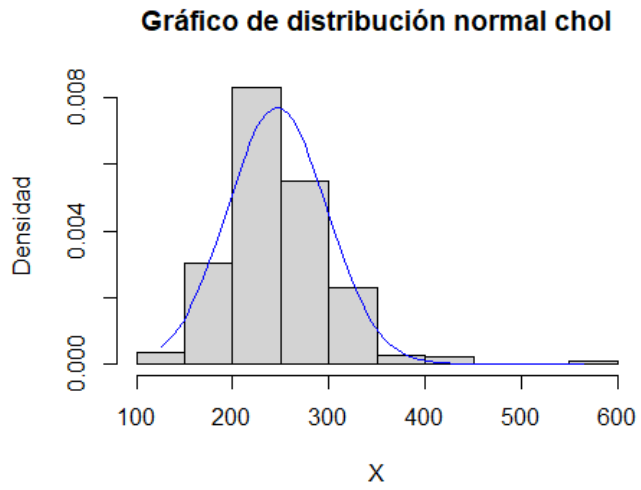
El test de Shapiro-Wilk se usa para contrastar si un conjunto de datos sigue una distribución normal o no. si el p-valor es menor al nivel de significancia α=0,05, la hipótesis nula se rechaza y se afirma que no sigue una distribución normal.

Podemos hacer unos gráficos superponiendo la distribución normal al histograma de frecuencias. Para ello hemos creado la función plotn() que lo hace.

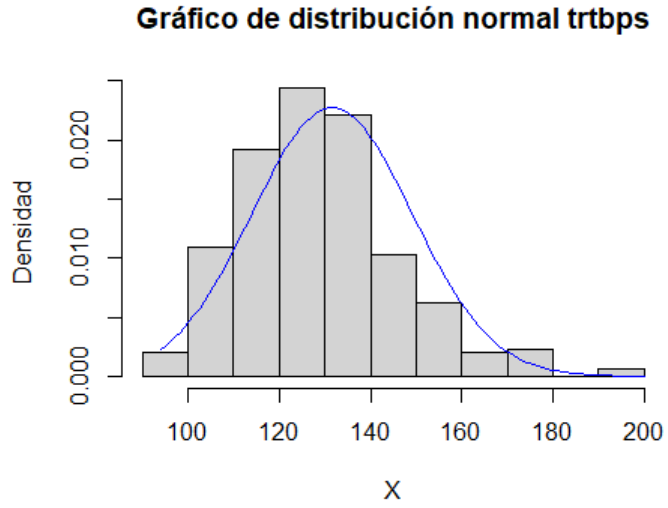
```
norm_test1 <- shapiro.test(heart$chol)
print(norm_test1)

##
##  Shapiro-Wilk normality test
##
## data:  heart$chol
## W = 0.94688, p-value = 5.365e-09

norm_test2 <- shapiro.test(heart$trtbps)
print(norm_test2)

##
##  Shapiro-Wilk normality test
##
## data:  heart$trtbps
## W = 0.96592, p-value = 1.458e-06

norm_test3 <- shapiro.test(heart$thalachh)
print(norm_test3)

##
##  Shapiro-Wilk normality test
##
## data:  heart$thalachh
## W = 0.97632, p-value = 6.621e-05

norm_test4 <- shapiro.test(heart$oldpeak)
print(norm_test4)

##
##  Shapiro-Wilk normality test
##
## data:  heart$oldpeak
## W = 0.84418, p-value < 2.2e-16
```

Gráfico en el que se superpone la línea de distribución normal.

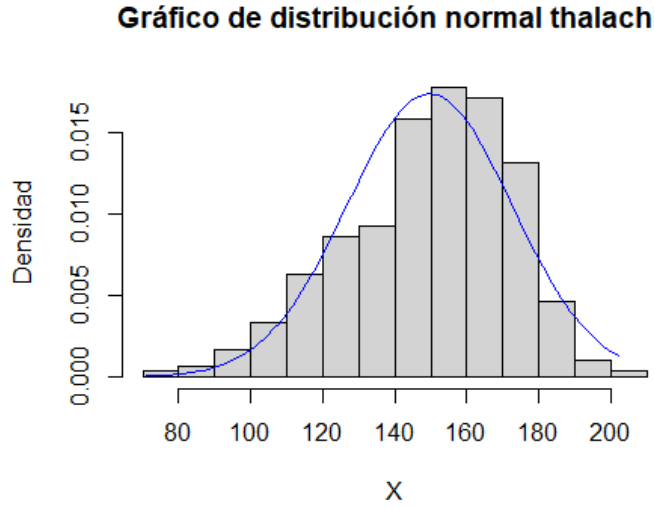
```
plotn <- function(x,main="Histograma de frecuencias \ny distribución normal",
                  xlab="X",ylab="Densidad") {
                  min <- min(x)
                  max <- max(x)
                  media <- mean(x)
                  dt <- sd(x)
                  hist(x,freq=F,main=main,xlab=xlab,ylab=ylab)
                  curve(dnorm(x,media,dt), min, max,add = T,col="blue")
}

plotn(heart$chol,main="Gráfico de distribución normal chol")
```



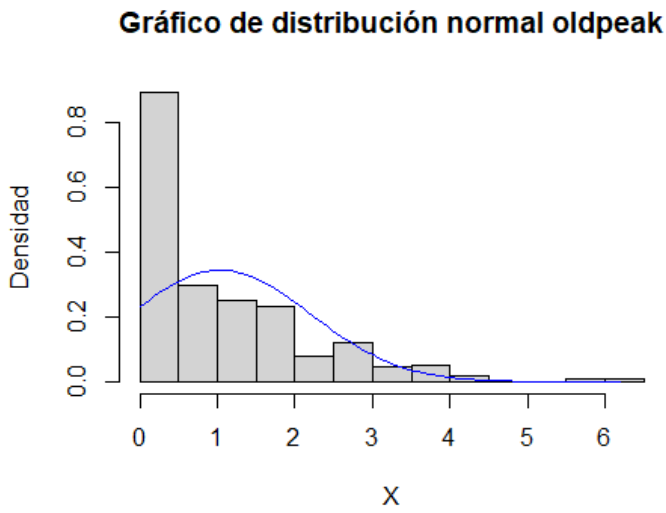

```
plotn(heart$trtbps,main="Gráfico de distribución normal trtbps")
```




```
plotn(heart$thalachh,main="Gráfico de distribución normal thalachh")
```



```
plotn(heart$oldpeak,main="Gráfico de distribución normal oldpeak")
```



Test de homogeneidad: Levene Test.

Evalúa la igualdad de las varianzas para una variable calculada para dos o más grupos. Algunos procedimientos estadísticos comunes asumen que las varianzas de las poblaciones de las que se extraen diferentes muestras son iguales. La prueba de Levene evalúa este supuesto.

Se pone a prueba la hipótesis nula de que las varianzas poblacionales son iguales. Si el P-valor resultante de la prueba de Levene es inferior a un cierto nivel de significación (0.05), la hipótesis nula de igualdad de varianzas se rechaza y se concluye que hay una diferencia entre las variaciones en la población.

Se va a evaluar la homogeneidad de la variable "`trtbps`".

Obtenemos un p-valor superior a 0,05, y podemos aceptar la hipótesis de que las varianzas de ambas muestras son homogéneas.

```
library(car)
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.1.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
leveneTest(y = heart$trtbps,group = heart$age, center = "median")
```

```
## Warning in leveneTest.default(y = heart$trtbps, group = heart$age, center =
## "median"): heart$age coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group  40  0.9174 0.6162
##       262
```

```
leveneTest(y = heart$trtbps,group = heart$sex, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   1  1.3593 0.2446
##       301
```

```
leveneTest(y = heart$trtbps,group = heart$cp , center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   3  1.4061 0.2411
##       299
```

```
leveneTest(y = heart$trtbps,group = heart$fbs , center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   1  1.0062 0.3166
##       301
```

```
leveneTest(y = heart$trtbps,group = heart$restecg , center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   2  0.7079 0.4935
##       300
```

```
leveneTest(y = heart$trtbps,group = heart$exng , center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   1  1.2021 0.2738
##       301
```

```
leveneTest(y = heart$trtbps,group = heart$slp , center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   2   1.191 0.3053
##       300
```

```
leveneTest(y = heart$trtbps,group = heart$caa , center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   4  0.9349  0.444
##       298
```

```
leveneTest(y = heart$trtbps,group = heart$thall, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   3  2.0058 0.1132
##       299
```

Test de homogeneidad de Fligner-Killeen:

Es un test no paramétrico que compara las varianzas basándose en la mediana. Es también una alternativa cuando no se cumple la condición de normalidad en las muestras.

Vamos a ver su aplicación para las variables “trtbps”, “chol”, “thalachh”, “oldpeak”, y “age”, con la variable “output”, que es la variable que nos indica si tenemos posibilidad o no de tener una enfermedad de corazón.

```
fligner.test(trtbps ~ output, data = heart)$p.value
```

```
## [1] 0.2423262
```

```
fligner.test(chol~ output, data = heart)$p.value
```

```
## [1] 0.4252618
```

```
fligner.test(thalachh ~ output, data = heart)$p.value
```

```
## [1] 0.02023875
```

```
fligner.test(oldpeak ~ output, data = heart)$p.value
```

```
## [1] 1.777207e-08
```

```
fligner.test(age  ~ output, data = heart)$p.value
```

```
## [1] 0.006898429
```

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.

Correlaciones entre variables numéricas, utilizando la función "cor", y el coeficiente de correlación Pearson. **Correlación entre las msimas variables utilizando "rquery".**

Usando la función "rquery.cormat" que nos muestra dos tablas y un gráfico:

- r : La tabla de coeficientes de correlación
- p : Tabla de p-valores correspondientes a los niveles de significancia de las correlaciones
- Una representación de la matriz de correlación en la que los coeficientes se reemplazan por símbolos de acuerdo con la fuerza de la dependencia. Las correlaciones negativas están en color azul y las positivas en color rojo.

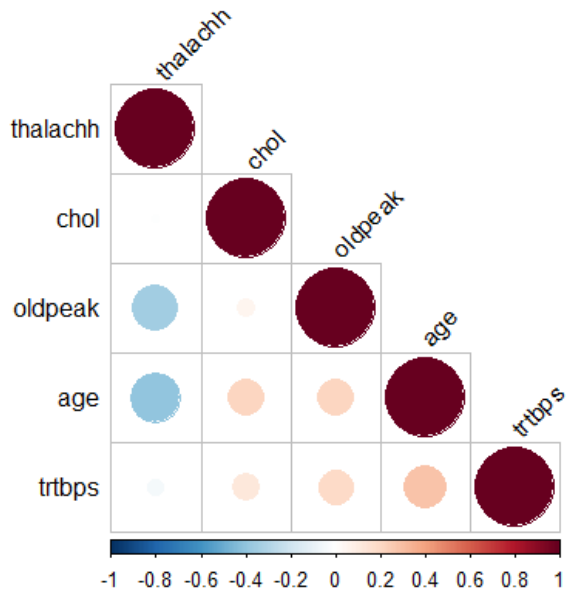
```
library(dplyr)
source("http://www.sthda.com/upload/rquery_cormat.r")

at_num <- select_if(heart,is.numeric)

cor(at_num , method="pearson")
```

```
##                 age      trtbps         chol     thalachh     oldpeak
## age       1.0000000  0.27935091  0.213677957 -0.398521938  0.21001257
## trtbps    0.2793509  1.00000000  0.123174207 -0.046697728  0.19321647
## chol      0.2136780  0.12317421  1.000000000 -0.009939839  0.05395192
## thalachh -0.3985219 -0.04669773 -0.009939839  1.000000000 -0.34418695
## oldpeak   0.2100126  0.19321647  0.053951920 -0.344186948  1.00000000
```

```
rquery.cormat(at_num )
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```




```
## $r
##          thalachh  chol oldpeak  age trtbps
## thalachh        1
## chol      -0.0099     1
## oldpeak     -0.34 0.054       1
## age          -0.4  0.21    0.21    1
## trtbps     -0.047  0.12    0.19 0.28      1
##
## $p
##          thalachh    chol oldpeak     age trtbps
## thalachh        0
## chol         0.86       0
## oldpeak   7.5e-10    0.35       0
## age       5.6e-13 0.00018 0.00023       0
## trtbps       0.42   0.032 0.00072 7.8e-07      0
##
## $sym
##          thalachh chol oldpeak age trtbps
## thalachh 1
## chol              1
## oldpeak  .             1
## age      .                    1
## trtbps                            1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

Como se aprecia los resultados de la matriz, se puede determinar que no hay una correlación lo suficientemente fuerte como para determinar que existe una correlación importante. Hay algunas correlaciones medias negativas entre las variables “age” – “thalachh”, “oldpeak” – “thalachh”. Y también correlaciones medias positivas entre las variables “trtbps” – “age”.

Modelo de regresión lineal

Una vez tenemos la correlación entre las variables numéricas, se puede ver por medio de regresión lineal la relación que tienen entre ellas. La relación se hará con la variable “age” que fue con la que mejor valor de correlación tienen las demás.

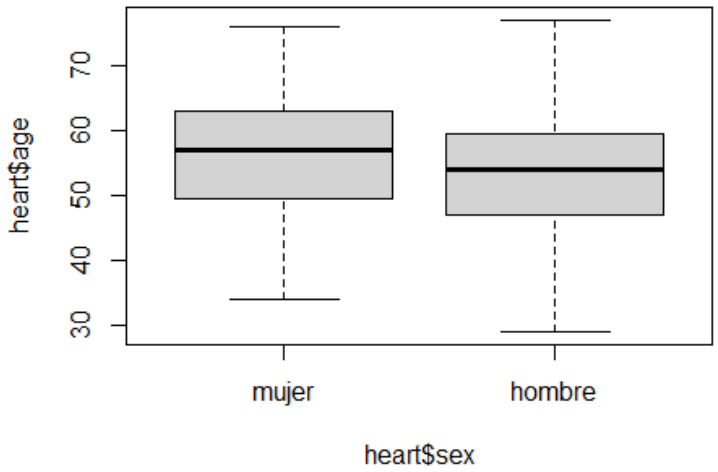
```
lr1<- lm(trtbps~age,data=heart)
summary(lr1)$adj.r.squared
```

```
## [1] 0.07497393

lr2 <-lm(chol ~ age,data=heart)
summary(lr2)$adj.r.squared

## [1] 0.0424877

lr3 <-lm(thalachh ~ age,data=heart)
summary(lr3)$adj.r.squared

## [1] 0.1560251

lr4 <-lm(oldpeak ~ age,data=heart)
summary(lr4)$adj.r.squared

## [1] 0.04092955
```

El coeficiente de determinación que se conoce más comúnmente como R-cuadrado (o R2), evalúa la fuerza de la relación lineal entre dos variables. Como vemos en los resultados, el coeficiente es muy bajo para todos los casos, lo que afirma que la variable dependiente no es predicha por la variable independiente.

T.test

```
boxplot(heart$age ~ heart$sex)
```

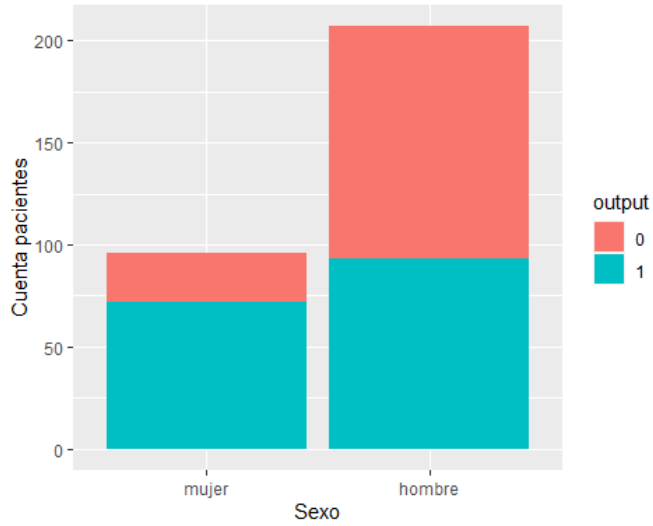


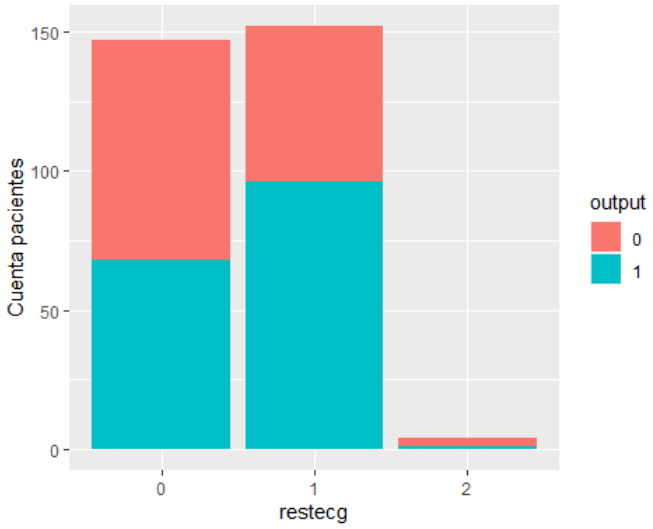

```
t.test(age ~ sex, data=heart)$p.value

## [1] 0.09463999
```

5.- Conclusiones

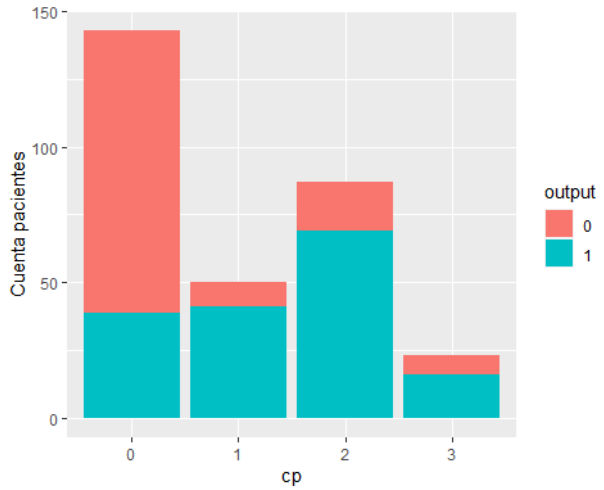
Para finalizar el trabajo, vamos a ilustrar gráficamente la relación entre las variables del sexo, resultados electrocardiográficos en reposo, tipo de dolor en el pecho, tasa de talasemia con la variable que indica si el paciente posee enfermedad de corazón.

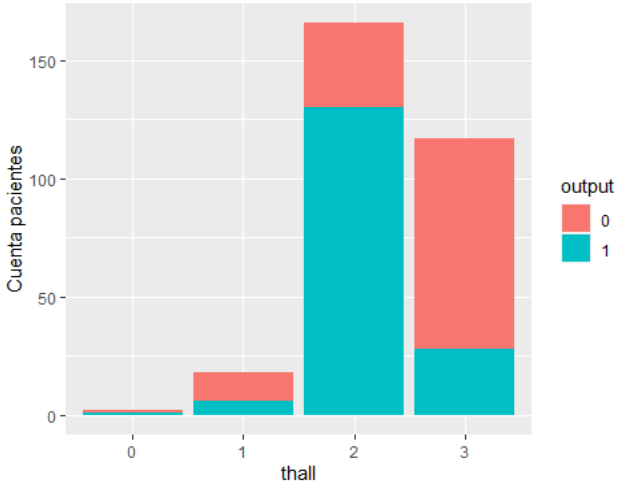
```
library(ggplot2)

ggplot(data = heart,aes(x=sex,fill=output)) + geom_bar() + xlab("Sexo") + ylab("Cuenta pacientes")

ggplot(data = heart,aes(x=restecg,fill=output)) + geom_bar() + xlab("restecg")+ ylab("Cuenta pacientes"
)
```






```
ggplot(data = heart,aes(x=cp,fill=output)) + geom_bar() + xlab("cp")+ ylab("Cuenta pacientes")

ggplot(data = heart,aes(x=thall,fill=output)) + geom_bar() + xlab("thall")+ ylab("Cuenta pacientes")
```





Se concluye con que:

- Los hombres por lo general sufren menos de enfermedades al corazón que las mujeres.
- Casi la mitad de los resultados electrocardiográficos en reposo cuando son normales, tienen enfermedades de corazón.
- Casi 2/3 de los resultados electrocardiográficos en reposo cuando hay anomalía de la onda ST-T, tienen enfermedades de corazón.

- Casi la totalidad de los resultados electrocardiográficos en reposo cuando hay hipertrofia ventricular, tienen enfermedades de corazón.
- 2/3 de los resultados de tipos de dolores de pecho cuando son angina típica, no poseen enfermedades de corazón. El resto de los dolores de pecho, si es muy probable de padecer enfermedad.
- 2/3 de la tasa de talasemia cuando tiene defecto reversible, no posee enfermedad de corazón.

En el resumen de lo estudiado, afirmamos que:

- El conjunto de datos no tenía valores nulos ni elementos vacíos. Sí que hay valores extremos, pero que no se han eliminado para ver su influencia en las enfermedades de corazón.
- El estudio de correlación entre variables numéricas muestra que no hay una correlación importante entre ellas.
- Las variables no tienen una distribución normal, excepto la variable que caracteriza al colesterol, que sigue cierta normalidad.
- Las variables “trtpbs” y “chol” tienen el p-valor superior a 0,05 y podemos aceptar la hipótesis de que las varianzas de ambas muestras son homogéneas. Las demás variables no lo superan.
- No se obtienen resultados buenos para los diferentes modelos de regresión lineal vistos.

6.- Código

El código se encuentra en:

HTTPS://GITHUB.COM/ADRIANGLEZ77/LIMPIEZA-Y-AN-LISIS-DE-DATOS.GIT

7.- Vídeo

https://drive.google.com/file/d/1Zynxgw5TnumLRb7ds8cqOgwpNQd_mJ3c/view?usp=share_link

Contribuciones

Contribuciones	Firma
Investigación previa	Adrián González González
Redacción de las respuestas	Adrián González González
Desarrollo del código	Adrián González González
Participación en el vídeo	Adrián González González