



Universitat Oberta
de Catalunya

M2.851 Tipología y ciclo de vida de los datos (aula 2)

Semestre 2022.2, PRA 02

Angel A. Urbina & Almudena Caballero Manzanar

5 de junio 2022

Carga de Librerías

Descarga datos desde GitHub

```
# Directorio de trabajo actual
Directorio <- getwd()

# Lectura de Datos
library(readr)
heart<-read.csv("https://raw.githubusercontent.com/UOCACM/PAC2/main/DATASET/heart.csv")

# Visualización Datos descargados
kable(head(heart, 3)) %>%
  kable_styling(full_width = FALSE)
```

Práctica 2 (25% nota final)

Presentación

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1

Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace al repositorio Git donde se encuentren las soluciones, incluyendo los nombres de los componentes del equipo. Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github.

Aunque no se trata del mismo enunciado ni de soluciones que obtuvieron la máxima nota, los siguientes ejemplos de ediciones anteriores os pueden servir como guía:

- Ejemplo: <https://github.com/Bengis/nba-gap-cleaning>
- Ejemplo complejo (archivo adjunto).

Además, se debe entregar un vídeo explicativo de la práctica, donde ambos integrantes del equipo expliquen con sus propias palabras el desarrollo de la práctica, basándose en las preguntas del enunciado para justificar y explicar el código desarrollado. Este vídeo se deberá entregar a través de un enlace a Google Drive que se deberá proporcionar junto con enlace al repositorio Git.

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.

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 continuar estudiando 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.

Descripción de la Práctica a realizar

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>).

Algunos ejemplos de dataset con los que podéis trabajar son:

- Red Wine Quality (<https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009>)
- Titanic: Machine Learning from Disaster (<https://www.kaggle.com/c/titanic>)

El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en esta competición.

Importante: si se elige un dataset diferente de los propuestos 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.

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset. (Puntuación 0.5 ptos)

Verificamos la estructura del juego de datos principal. Vemos el número de columnas que tenemos y ejemplos de los contenidos de las filas, así como un primer análisis de los valores (mínimo, máximo, media, mediana, ...) de cada una de las variables.

```
# Estructura de los datos
structure <- 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 ...
## $ trestbps: 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 ...
## $ thalach : int 150 187 172 178 163 148 153 173 162 174 ...
## $ exang : 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 ...
## $ slope : int 0 0 2 2 2 1 1 2 2 2 ...
## $ ca : int 0 0 0 0 0 0 0 0 0 0 ...
## $ thal : int 1 2 2 2 2 1 2 3 3 2 ...
## $ target : int 1 1 1 1 1 1 1 1 1 1 ...
```

Tipos de Datos

```
sapply(heart, class)
```

```
##      age      sex      cp trestbps      chol      fbs  restecg  thalach
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##      exang  oldpeak      slope      ca      thal      target
## "integer" "numeric" "integer" "integer" "integer" "integer"
```

```
summary(heart)
```

```
##      age      sex      cp      trestbps
## 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      thalach
## 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
##      exang      oldpeak      slope      ca
## 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
##      thal      target
## 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
```

Tamaño Datos

```
dim(heart)
```

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

Resumen Datos

Nuestro DataSet esta originalmente en:

<https://www.kaggle.com/datasets/zhaoyingzhu/heartcsv>

Son de caracter publico

Vemos que tenemos **14** variables y **303** registros

Revisamos la descripción de las variables contenidas en el fichero y si los tipos de variables se corresponden con las que hemos cargado. Las organizamos lógicamente para darles sentido y construimos un pequeño diccionario de datos utilizando la documentación auxiliar.

- **age** col_double(), [Edad Paciente]
- **sex** col_double(), [Sexo Paciente] 0 = female 1 = male
- **cp** col_double(), [Tipo de Dolor de Pecho] 1='typical angina' 2 = 'atypical angina' 3 = 'non-anginal pain' 4 = 'asymptomatic'
- **trestbps** col_double(), [resting_blood_pressure]
- **chol** col_double(), [cholesterol]
- **fbs** col_double(), [fasting_blood_sugar] 0='lower than 120mg/ml' 1='greater than 120mg/ml'
- **restecg** col_double(), [rest_ecg] 0 = 'normal' 1 = 'ST-T wave abnormality' 2 = 'left ventricular hypertrophy'
- **thalach** col_double(), [max_heart_rate_achieved]
- **exang** col_double(), [exercise_induced_angina] 0 = 'no' 1 = 'yes'
- **oldpeak** col_double(), [st_depression]
- **slope** col_double(), [st_slope] 1 = 'upsloping' 2 = 'flat' 3 = 'downsloping'
- **ca** col_double(), [num_major_vessels]
- **thal** col_double(), [thalassemia] 1 = 'normal' 2 = 'fixed defect' 3 = 'reversible defect'
- **target** col_double() [target] Yes No

Resumen DataSet

- Numero de variables 15
- Numero de observaciones 303
- Celdas perdidas 6 (0.1%)
- Filas duplicadas 0 (0.0%)

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

Este Data set recopila un conjunto de datos clinicos cuyo objetivo es permitir intentar identificar posibles factores que afectan a la presencia de infartos en personas.

Es importante ya que se trata de un conjunto de datos reales que nos permitir construir modelos para la predicción de este tipo de situaciones que son una de las causas mas frecuentes de fallecimientos en las personas.

2. Integración y selección de los datos de interés a analizar. (Puntuación 0.5 pts)

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

Procedemos a construir vectores que nos permitan mejorar la identificación de las variables

```
sex <- c('female',
        'male')
cp <- c('typical angina',
        'atypical angina',
        'non-anginal pain',
        'asymptomatic')
fbs <- c('lower than 120mg/ml',
        'greater than 120mg/ml')
rest_ecg <- c('normal',
              'ST-T wave abnormality',
```

```

        'left ventiricular hypertrophy')
exercise_induced_angina <- c( 'yes',
                             'no')

st_slope <- c('upsloping',
              'flat',
              'downsloping')
thalassemia <- c('normal',
                 'fixed defect',
                 'reversable defect')
infarto <- c('Yes',
             'No')

```

Construcción Dataset Factorizado

```

# Cambio valores
Heart_fact <- heart %>%
  mutate(sex = case_when(sex == 0 ~ 'female',
                         sex == 1 ~ 'male'),
         cp = case_when(cp == 0 ~ 'typical angina',
                        cp == 1 ~ 'atypical angina',
                        cp == 2 ~ 'non-anginal pain',
                        cp == 3 ~ 'asymptomatic'
                        ),
         fbs = case_when(fbs == 0 ~ 'lower than 120mg/ml',
                        fbs == 1 ~ 'greater than 120mg/ml'
                        ),
         restecg = case_when(restecg == 0 ~ 'normal',
                             restecg == 1 ~ 'ST-T wave abnormality',
                             restecg == 2 ~ 'left ventiricular hypertrophy'
                             ),
         exang = case_when(exang == 0 ~ 'no',
                           exang == 1 ~ 'yes'
                           ),
         target = case_when(target == 0 ~ FALSE,
                             target == 1 ~ TRUE
                             ),
         slope = case_when(slope == 0 ~ 'upsloping',
                           slope == 1 ~ 'flat',
                           slope == 2 ~ 'downsloping'
                           ),
         ca = case_when(ca == 0 ~ '0',
                        ca == 1 ~ '1',
                        ca == 2 ~ '2',
                        ca == 3 ~ '3',
                        ca == 4 ~ '4'
                        ),
         thal = case_when(thal == 1 ~ 'normal',
                          thal == 2 ~ 'fixed defect',
                          thal == 3 ~ 'reversable defect',
                          thal == 0 ~ 'Sin Datos'
                          ))

```

```
# Factorización
Heart_fact <- Heart_fact %>%
  mutate(sex = relevel(as.factor(sex), 'female', 'male'),
         cp = relevel(as.factor(cp), 'typical angina', 'atypical angina', 'non-anginal pain', 'asymptomatic'),
         fbs = relevel(as.factor(fbs), 'lower than 120mg/ml', 'greater than 120mg/ml'),
         restecg = relevel(as.factor(restecg), 'normal', 'ST-T wave abnormality', 'left ventricular hypertrophy or enlargement'),
         exang = relevel(as.factor(exang), 'no', 'yes'),
         slope = relevel(as.factor(slope), 'upsloping', 'flat', 'downsloping'),
         thal = relevel(as.factor(thal), 'normal', 'fixed defect', 'reversible defect', 'Sin Datos'),
         ca = relevel(as.factor(ca), '0', '1', '2', '3', '4'),
         target= relevel(as.factor(target), TRUE, FALSE)
  )
```

3. Limpieza de los datos. (Puntuación 2 pts)

3.1. ¿Los datos contienen ceros o elementos vacíos?

Comprobacion presencia de NA

```
datosNA <- colSums(is.na(Heart_fact))
print(datosNA)
```

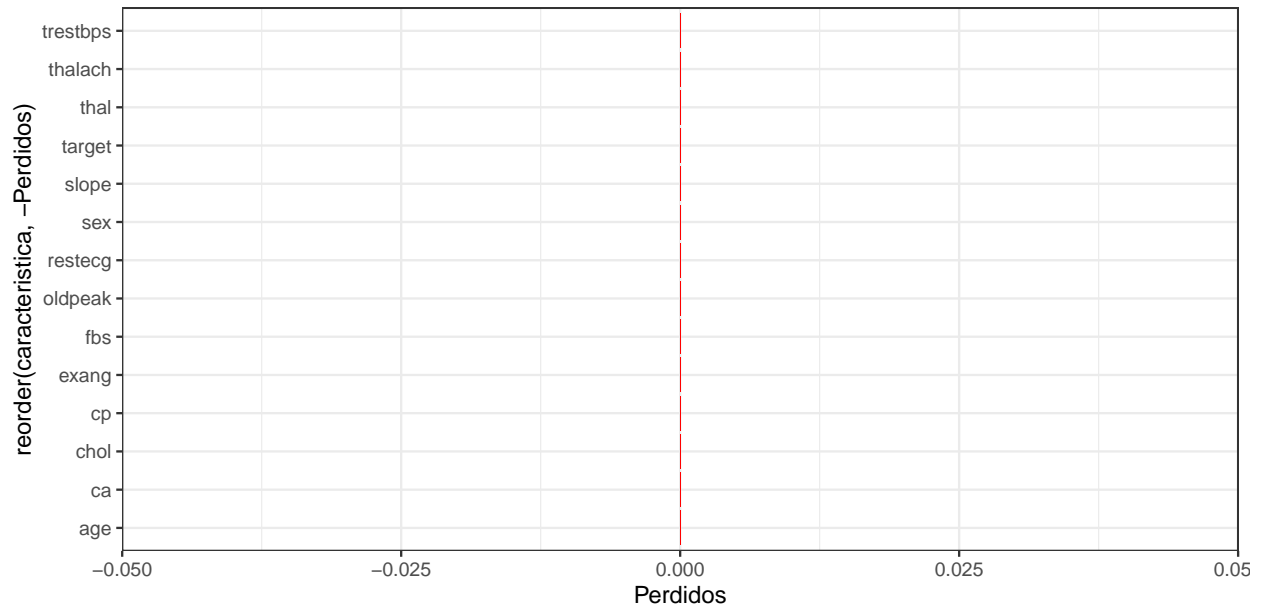
```
##      age      sex      cp trestbps      chol      fbs  restecg  thalach
##       0       0       0       0       0       0       0       0
##  exang  oldpeak    slope      ca      thal  target
##       0       0       0       0       0       0
```

No hay valores NA

Tabla resumen Valores perdidos

```
missing_values <- Heart_fact %>% summarize_each(funs(sum(is.na(.))/n()))

missing_values <- gather(missing_values, key="caracteristica", value="Perdidos")
missing_values %>%
  ggplot(aes(x=reorder(caracteristica,-Perdidos),y=Perdidos)) +
  geom_bar(stat="identity",fill="red")+
  coord_flip()+theme_bw()
```



No hay valores perdidos

Gestiona cada uno de estos casos.

Se trata de una fuente de datos que no contiene valores NA ni valores perdidos ya que procede de datos de una investigación clínica. (Asumimos que por tanto que a los valores publicados Kaggle que hemos cogido ya se han realizado los pasos indicados de limpieza)

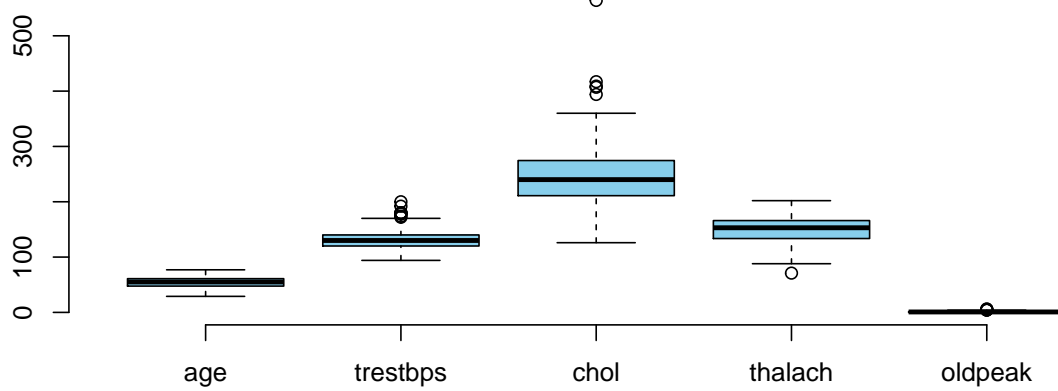
3.2. Identifica y gestiona los valores extremos.

Primero identificamos las columnas numéricas

```
Numericos <- Heart_fact %>%
  dplyr::select(where(is.numeric))
```

Dibujamos Boxplots

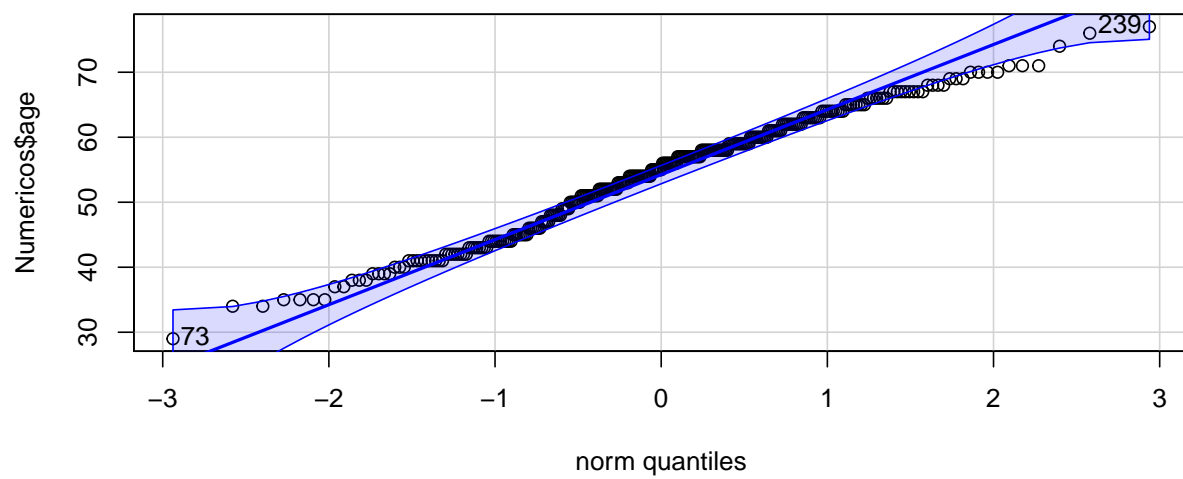
```
g_caja<-boxplot(Numericos, col="skyblue", frame.plot=F)
```

Verificación Normalidad

Variable age

```
library(car)
qqPlot(Numericos$age)
```



```
## [1] 73 239
```

```
shapiro.test(Numericos$age)
```

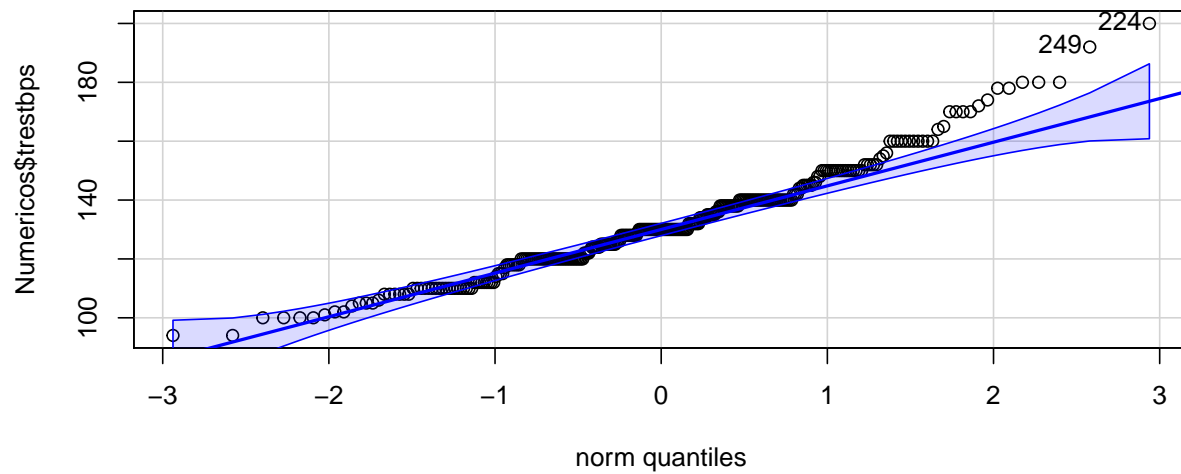
```
##
```

```
## Shapiro-Wilk normality test
```

```
##
## data: Numericos$age
## W = 0.98637, p-value = 0.005798
```

Variable trestbps

```
qqPlot(Numericos$trestbps)
```



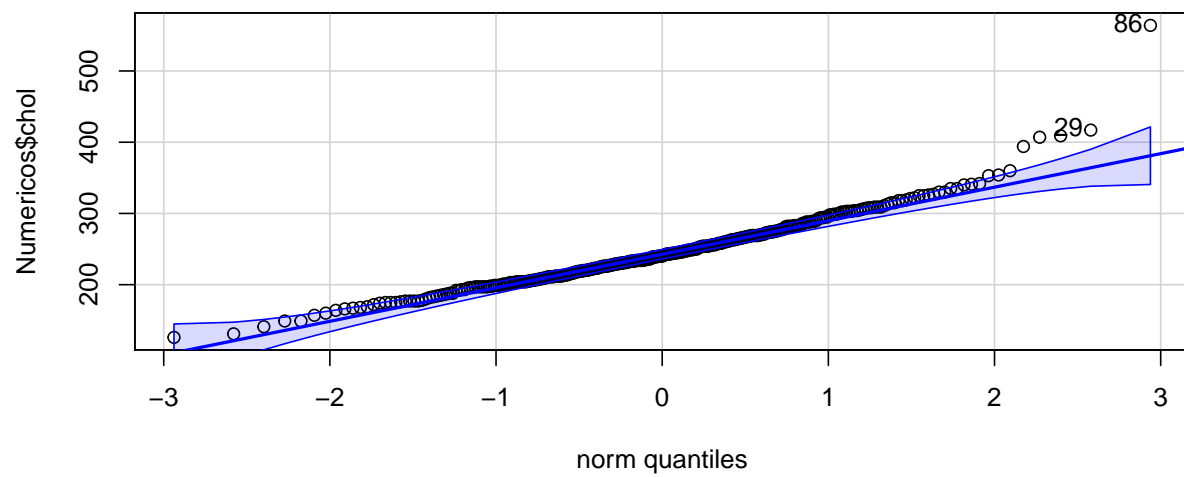
```
## [1] 224 249
```

```
shapiro.test(Numericos$trestbps)
```

```
##
## Shapiro-Wilk normality test
##
## data: Numericos$trestbps
## W = 0.96592, p-value = 1.458e-06
```

Variable chol

```
qqPlot(Numericos$chol)
```



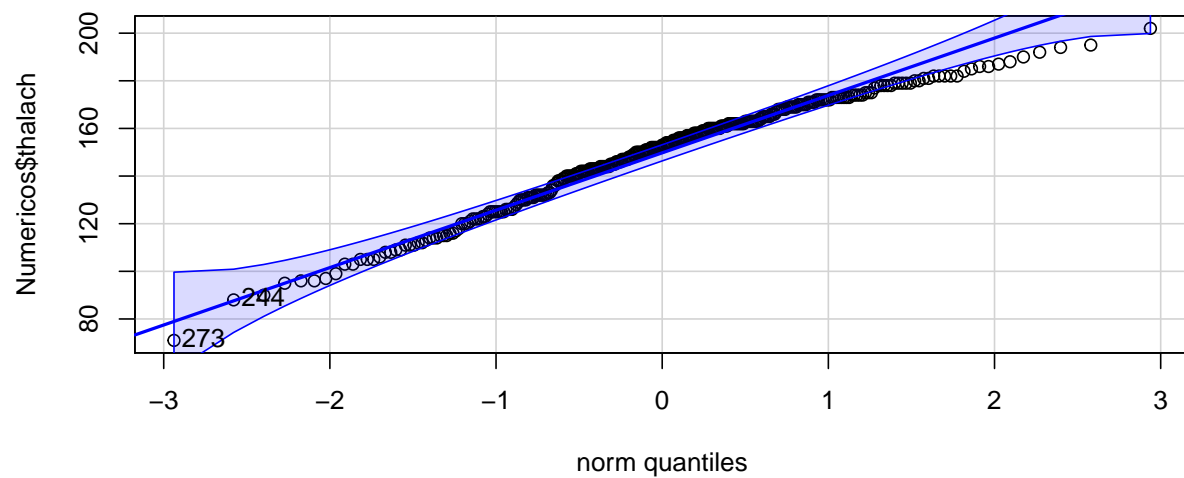
```
## [1] 86 29
```

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

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

Variable thalach

```
qqPlot(Numericos$thalach)
```



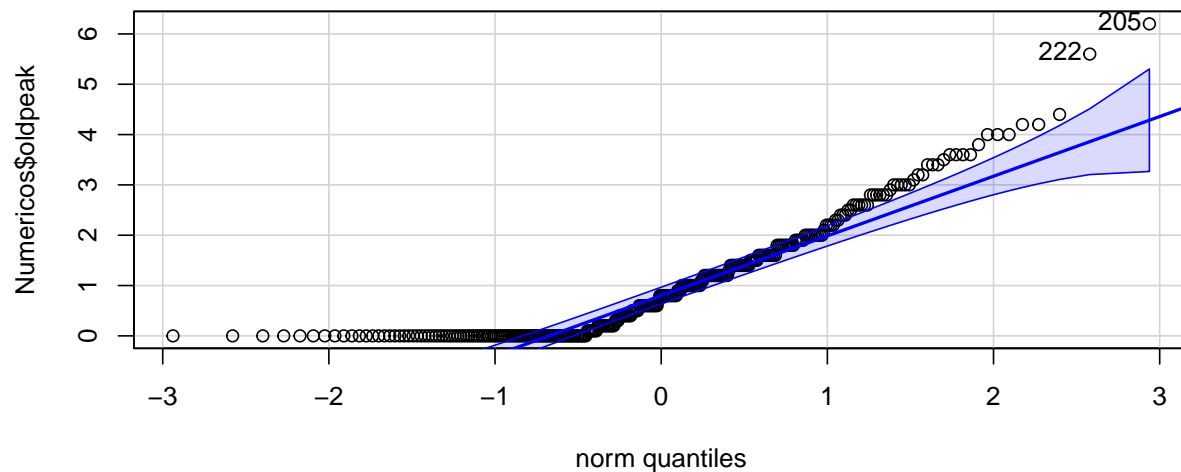
```
## [1] 273 244
```

```
shapiro.test(Numericos$thalach)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: Numericos$thalach  
## W = 0.97632, p-value = 6.621e-05
```

Variable oldpeak

```
qqPlot(Numericos$oldpeak)
```



```
## [1] 205 222
```

```
shapiro.test(Numericos$oldpeak)
```

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

Identificacion valores extremos

Variable Age

```
out <- boxplot.stats(Numericos$age)$out  
# valores  
out
```

```
## integer(0)
```

```
out_ind <- which(Numericos$age %in% c(out))  
# indices  
out_ind
```

```
## integer(0)
```

No hay outliers de Age

Variable trestbps

```
out <- boxplot.stats(Numericos$trestbps)$out  
# valores  
out
```

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

```
out_ind <- which(Numericos$trestbps %in% c(out))  
# valores  
out_ind
```

```
## [1] 9 102 111 204 224 242 249 261 267
```

Variable chol

```
out <- boxplot.stats(Numericos$chol)$out  
# valores  
out
```

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

```
out_ind <- which(Numericos$chol %in% c(out))  
# valores  
out_ind
```

```
## [1] 29 86 97 221 247
```

Variable thalach

```
out <- boxplot.stats(Numericos$thalach)$out  
# valores  
out
```

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

```
out_ind <- which(Numericos$thalach %in% c(out))  
# valores  
out_ind
```

```
## [1] 273
```

Variable oldpeak

```
out <- boxplot.stats(Numericos$oldpeak)$out
# valores
out
```

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

```
out_ind <- which(Numericos$oldpeak %in% c(out))
# valores
out_ind
```

```
## [1] 102 205 222 251 292
```

4. Análisis de los datos. (Puntuación 2.5 pts)

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

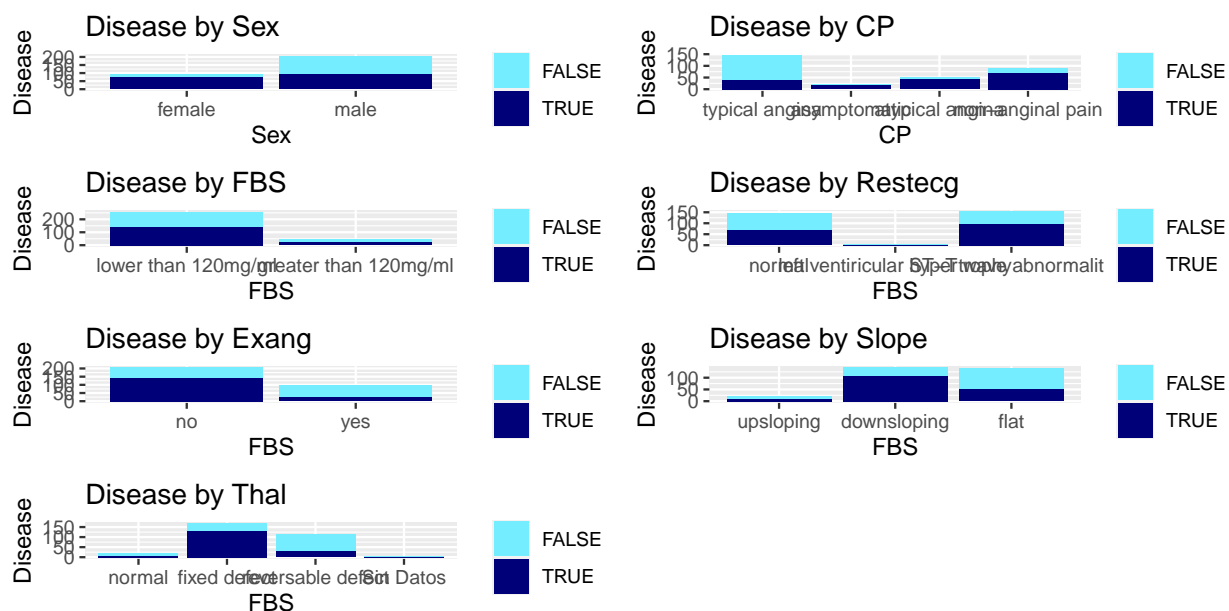
(p. e., si se van a comparar grupos de datos, ¿cuáles son estos grupos y qué tipo de análisis se van a aplicar?)

Nuestro objetivo es determinar la probabilidad de sufrir una enfermedad cardiaca. Para ello, veamos en primer lugar, la relación de algunas de las variables con dicha probabilidad.

```
grid.newpage()
```

```
plotbySex<-ggplot(Heart_fact,aes(sex,fill=target))+geom_bar() +labs(x="Sex", y="Disease")+ guides(fill=
plotbycp<-ggplot(Heart_fact,aes(cp,fill=target))+geom_bar() +labs(x="CP", y="Disease")+ guides(fill=gui
plotbyfbs<-ggplot(Heart_fact,aes(fbs,fill=target))+geom_bar() +labs(x="FBS", y="Disease")+ guides(fill=
plotbyreste<-ggplot(Heart_fact,aes(restecg,fill=target))+geom_bar() +labs(x="FBS", y="Disease")+ guides
plotbyexang<-ggplot(Heart_fact,aes(exang,fill=target))+geom_bar() +labs(x="FBS", y="Disease")+ guides(f
plotbyrslope<-ggplot(Heart_fact,aes(slope,fill=target))+geom_bar() +labs(x="FBS", y="Disease")+ guides(
plotbythal<-ggplot(Heart_fact,aes(thal,fill=target))+geom_bar() +labs(x="FBS", y="Disease")+ guides(fill
```

```
grid.arrange(plotbySex,plotbycp,plotbyfbs,plotbyreste,plotbyexang,plotbyrslope,plotbythal,ncol=2)
```



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

Revisamos si las variables numéricas están normalizadas con el test de Shapiro-Wilk

```
shapiro.test(Heart_fact$trestbps)

##
##  Shapiro-Wilk normality test
##
## data:  Heart_fact$trestbps
## W = 0.96592, p-value = 1.458e-06
```

```
shapiro.test(Heart_fact$chol)

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

```
shapiro.test(Heart_fact$thalach)

##
##  Shapiro-Wilk normality test
##
## data:  Heart_fact$thalach
## W = 0.97632, p-value = 6.621e-05
```

```
shapiro.test(Heart_fact$oldpeak)

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

Puesto que en todos los casos el p-valor es inferior a 0.05 podemos rechazar la hipótesis nula y concluir que las variables no siguen una distribución normal.

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.

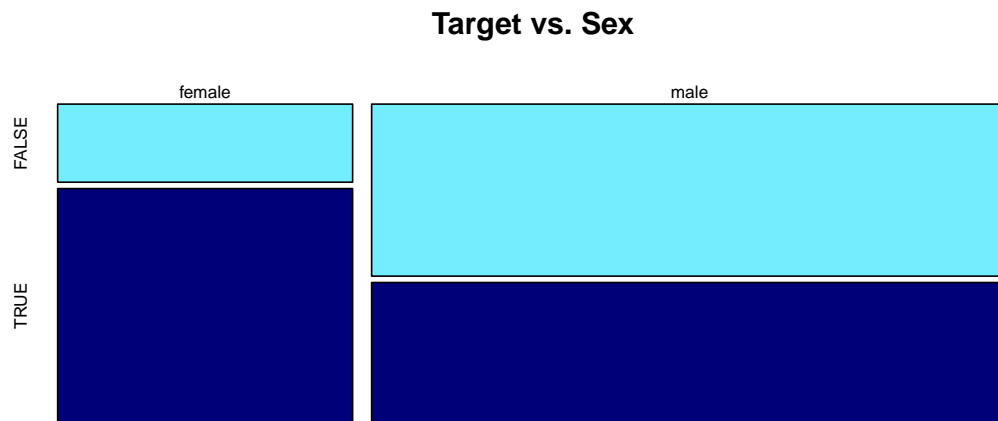
Veamos a continuación cómo se distribuye el target según las diferentes variables

```

tabla_sex <- table(Heart_fact$sex,Heart_fact$target)
tabla_cp <- table(Heart_fact$cp,Heart_fact$target)
tabla_fbs <- table(Heart_fact$fbs,Heart_fact$target)
tabla_restecg <- table(Heart_fact$restecg,Heart_fact$target)
tabla_slope <- table(Heart_fact$slope,Heart_fact$target)
tabla_thal <- table(Heart_fact$thal,Heart_fact$target)

grid.newpage()
plot(tabla_sex, col = c("#73EDFF","#000078"), main = "Target vs. Sex")

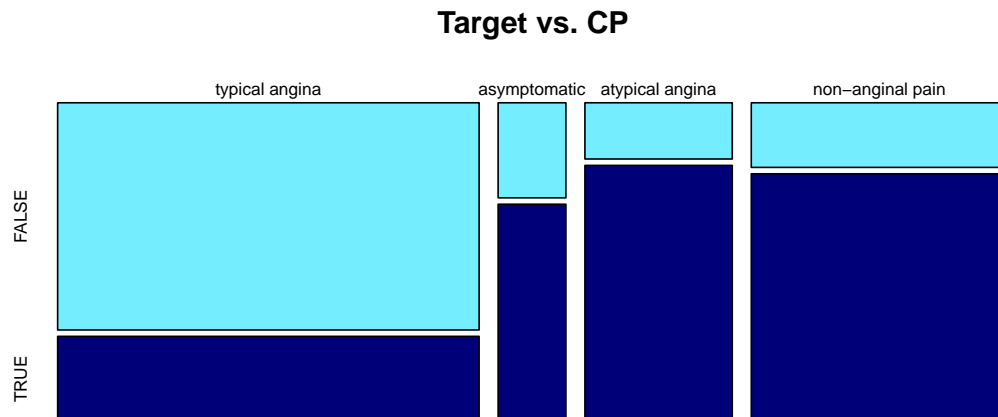
```



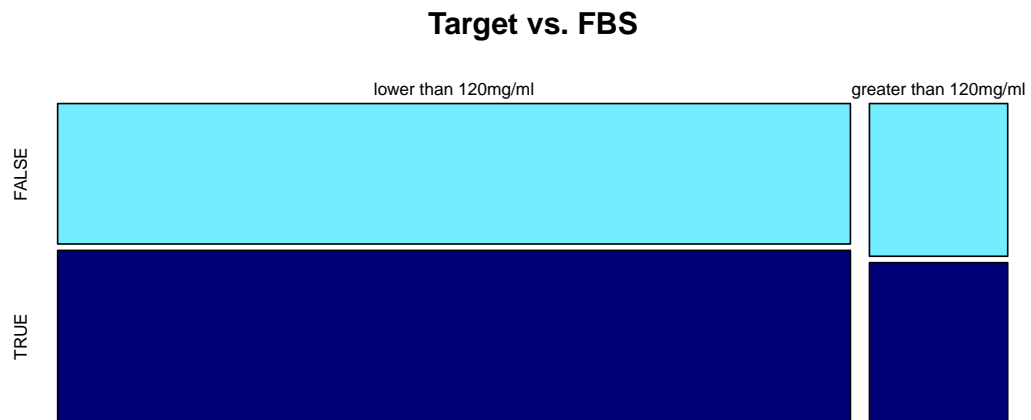
```

plot(tabla_cp, col = c("#73EDFF","#000078"), main = "Target vs. CP")

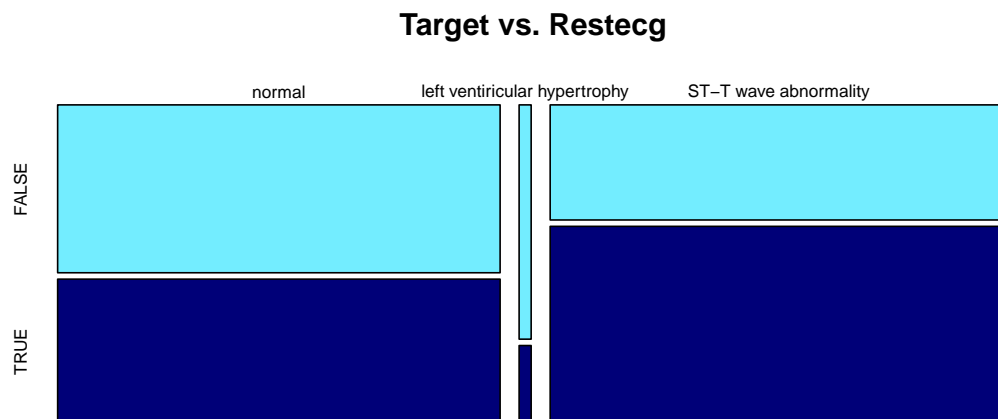
```




```
plot(tabla_fbs, col = c("#73EDFF", "#000078"), main = "Target vs. FBS")
```

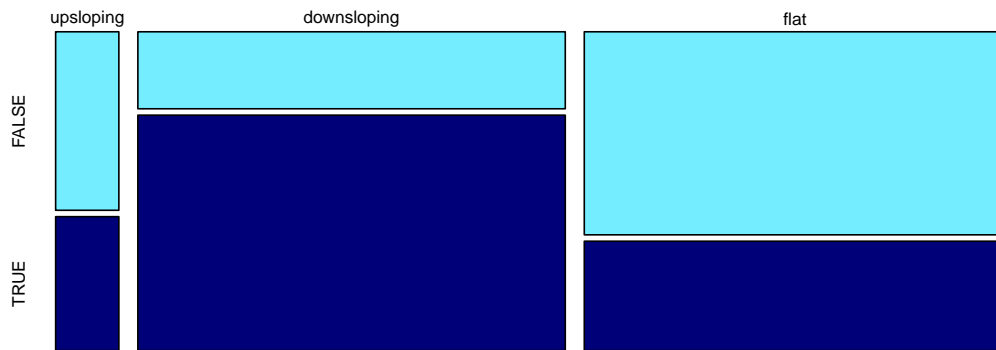


```
plot(tabla_restecg, col = c("#73EDFF", "#000078"), main = "Target vs. Restecg")
```



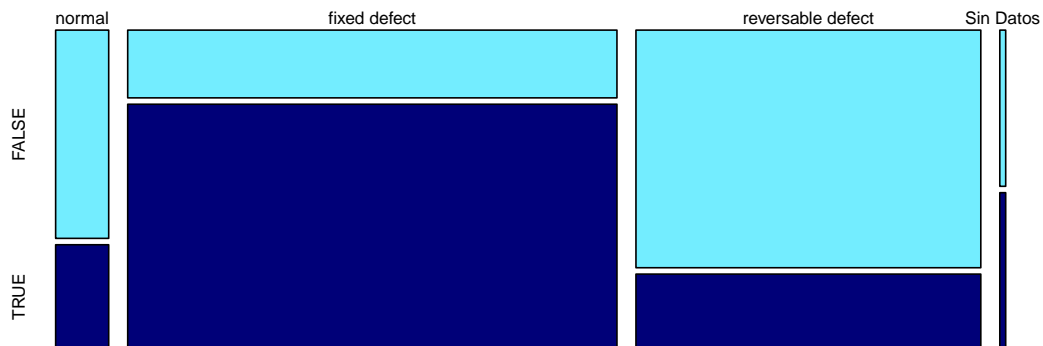
```
plot(tabla_slope, col = c("#73EDFF", "#000078"), main = "Target vs. Slope")
```

Target vs. Slope



```
plot(tabla_thal, col = c("#73EDFF", "#000078"), main = "Target vs. Thal")
```

Target vs. Thal



Vemos que en algunas de las variables, la distribución es considerablemente diferente. Así, por género, observamos que las mujeres tienen mayor probabilidad de sufrir enfermedades coronarias. Ocurre algo similar con la variable Thal, en la que el valor 'fixed defect' tiene un porcentaje mayor que el resto. Nos planteamos ahora estudiar la correlación entre algunas variables. Como vimos en el apartado anterior, nuestras variables no se ajustan a una distribución normal por lo que aplicaremos la correlación de Spearman.

```
cor.test(Heart_fact$chol, Heart_fact$trestbps, method = "spearman")
```

```
##  
## Spearman's rank correlation rho
```

```
##
## data: Heart_fact$chol and Heart_fact$trestbps
## S = 4049526, p-value = 0.02761
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1265616

cor.test(Heart_fact$oldpeak,Heart_fact$trestbps,method="spearman")

##
## Spearman's rank correlation rho
##
## data: Heart_fact$oldpeak and Heart_fact$trestbps
## S = 3921077, p-value = 0.007138
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1542667
```

No se aprecia correlación entre los pares de variables estudiados.

Veamos ahora la regresión logística teniendo en cuenta las dos variables anteriores en los que los porcentajes de enfermedad eran considerablemente diferentes según los valores, sex y thal:

```
chol_trest <- glm(target ~ sex + thal, data = Heart_fact,
                  family=binomial(logit))
summary(chol_trest)

##
## Call:
## glm(formula = target ~ sex + thal, family = binomial(logit),
##      data = Heart_fact)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8939  -0.7129   0.6033   0.7807   1.7287
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.1481    0.5836  -0.254  0.79963
## sexmale        -0.5796    0.3190  -1.817  0.06925 .
## thalfixed defect  1.7596    0.5461   3.222  0.00127 **
## thalreversible defect -0.5124    0.5472  -0.936  0.34904
## thalSin Datos    0.4379    1.5208   0.288  0.77339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 417.64  on 302  degrees of freedom
## Residual deviance: 324.75  on 298  degrees of freedom
## AIC: 334.75
```

```
##
## Number of Fisher Scoring iterations: 4
```

Observamos que, a excepción del valor ‘fixed defect’ de la variable thal, los valores no son significativos.

El valor AIC es 334.75, si comparasemos varios modelos, deberemos considerar aquel con un valor AIC inferior.

Si incorporamos más variables a dicha regresión obtenemos:

```
reg <- glm(target ~ sex + thal + chol + fbs + restecg + exang + slope + cp, data = Heart_fact,
            family=binomial(logit))
summary(reg)
```

```
##
## Call:
## glm(formula = target ~ sex + thal + chol + fbs + restecg + exang +
##      slope + cp, family = binomial(logit), data = Heart_fact)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.5392  -0.5206   0.2008   0.5458   2.4488
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.022043    1.224730   0.835 0.403996
## sexmale          -1.320573    0.437816  -3.016 0.002559 **
## thalfixed defect    0.561489    0.651207   0.862 0.388563
## thalreversible defect -1.069788    0.642116  -1.666 0.095706 .
## thalSin Datos     -0.624182    2.726777  -0.229 0.818940
## chol             -0.003689    0.003388  -1.089 0.276106
## fbsgreater than 120mg/ml -0.195468    0.478330  -0.409 0.682800
## restecgleft ventricular hypertrophy -1.135568    1.644606  -0.690 0.489892
## restecgST-T wave abnormality  0.605265    0.341735   1.771 0.076535 .
## exangyes         -0.822805    0.378263  -2.175 0.029614 *
## slopedownslowing    1.010243    0.655763   1.541 0.123424
## slopeflat         -0.390958    0.633141  -0.617 0.536911
## cpasymptomatic     2.057281    0.615630   3.342 0.000833 ***
## cpatypical angina    1.509028    0.497977   3.030 0.002443 **
## cpnon-anginal pain   1.868119    0.425702   4.388 1.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 417.64  on 302  degrees of freedom
## Residual deviance: 238.75  on 288  degrees of freedom
## AIC: 268.75
##
## Number of Fisher Scoring iterations: 5
```

Observamos que con la inclusión de más variables obtenemos mejores modelos ya que el nuevo AIC toma como valor 268.75

Continuamos con el estudio de las variables sex y thal, y nos planteamos si hay diferencias significativas entre los valores de la variables thal y el género. Para ello aplicamos el test chi cuadrado:

```
rel <- table(Heart_fact$sex,Heart_fact$thal)
print(rel)
```

```
##
##           normal fixed defect reversable defect Sin Datos
##   female         1         79             15         1
##   male          17         87            102         1
```

```
chisq.test(rel)
```

```
##
## Pearson's Chi-squared test
##
## data:  rel
## X-squared = 44.626, df = 3, p-value = 1.111e-09
```

Vemos que efectivamente hombres y mujeres tienen diferente distribución de la variable thal

5 Representación de los resultados a partir de tablas y gráficas. (Puntuación 2 pts) Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica.

6 Resolución del problema. (Puntuación 0.5 pts) A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

A partir del dataset considerado, hemos tratado los datos y hecho un estudio de cada una de las variables. A su vez, hemos relacionado cada una de éstas con la probabilidad de sufrir alguna enfermedad coronaria obteniendo datos interesantes con las variables sex y thal.

Hemos visto que la variable thal se distribuye de manera diferente según el género y que el valor 'fixed defect' toma un carácter significativo por lo que podría considerarse una variable importante a la hora de establecer futuras predicciones.

7 Código: (Puntuación 2 pts) Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos.

Si lo preferís, también podéis trabajar en Python.