Practica 2: Data Cleaning

Jose Luis Rivas Calduch y Mariano Jiménez Barca

30/11/2020

Indice de contenidos

1.- Descripción del Dataset

Este dataset recoge datos de pacientes reales que han sufrido un infarto de miocardio y que o bien han fallecido o bien han sobrevivido al cabo de un tiempo (recogido en la variable time).

Los datos que recoge el dataset nos informan de datos médicos en el momento del ataque y permite a priori crear modelos predictivos respecto a la probablidad de supervivencia de una persona tras un infarto de miocardio en función de sus datos analíticos.

Permitiría preguntas de tipo ¿Es más probable que sobreviva un paciente fumador a un ataque al corazón? ¿Es más probable que sobreviva una persona de sexo femenino? ¿y una persona con diabetes?

Permitiría generar un modelo predictivo que informara de cuáles son los pacientes más o menos probables de fallecer en función de una sere de condiciones: edad, concentración de creatinina en suero... y focalizarse más en este tipo de pacientes para tratar de aumentar la posibilidad de supervivencia.

Los datos los podemos encontrar aqui: https://www.kaggle.com/andrewmvd/heart-failure-clinical-data

Las variables del dataset son:

Carga del fichero.

```
# carga del fichero
hf <- read.table("../data/heart_failure_clinical_records_dataset.csv", header= TRUE, sep=",", dec="." )

head(hf)
```

```
##   age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 1  75       0                     582        0                20
## 2  55       0                    7861        0                38
## 3  65       0                     146        0                20
## 4  50       1                     111        0                20
## 5  65       1                     160        1                20
```

```
## 6  90       1                       47        0                40
##   high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 1                   1    265000             1.9          130   1       0    4
## 2                   0    263358             1.1          136   1       0    6
## 3                   0    162000             1.3          129   1       1    7
## 4                   0    210000             1.9          137   1       0    7
## 5                   0    327000             2.7          116   0       0    8
## 6                   1    204000             2.1          132   1       1    8
##   DEATH_EVENT
## 1           1
## 2           1
## 3           1
## 4           1
## 5           1
## 6           1
```

Descripción de las variables

```
str(hf)
```

```
## 'data.frame':    299 obs. of  13 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
##  $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ platelets               : num  265000 263358 162000 210000 327000 ...
##  $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
##  $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
##  $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
##  $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
##  $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

Resumen descriptivo de las variables

```
summary(hf)
```

```
##       age           anaemia       creatinine_phosphokinase    diabetes
##  Min.   :40.00   Min.   :0.0000   Min.   :  23.0           Min.   :0.0000
##  1st Qu.:51.00   1st Qu.:0.0000   1st Qu.: 116.5           1st Qu.:0.0000
##  Median :60.00   Median :0.0000   Median : 250.0           Median :0.0000
##  Mean   :60.83   Mean   :0.4314   Mean   : 581.8           Mean   :0.4181
##  3rd Qu.:70.00   3rd Qu.:1.0000   3rd Qu.: 582.0           3rd Qu.:1.0000
##  Max.   :95.00   Max.   :1.0000   Max.   :7861.0           Max.   :1.0000
##  ejection_fraction high_blood_pressure   platelets      serum_creatinine
##  Min.   :14.00     Min.   :0.0000      Min.   : 25100   Min.   :0.500
##  1st Qu.:30.00     1st Qu.:0.0000      1st Qu.:212500   1st Qu.:0.900
##  Median :38.00     Median :0.0000      Median :262000   Median :1.100
##  Mean   :38.08     Mean   :0.3512      Mean   :263358   Mean   :1.394
##  3rd Qu.:45.00     3rd Qu.:1.0000      3rd Qu.:303500   3rd Qu.:1.400
```

```
##  Max.   :80.00     Max.   :1.0000      Max.   :850000   Max.   :9.400
##   serum_sodium        sex             smoking            time
##  Min.   :113.0   Min.   :0.0000   Min.   :0.0000   Min.   :  4.0
##  1st Qu.:134.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 73.0
##  Median :137.0   Median :1.0000   Median :0.0000   Median :115.0
##  Mean   :136.6   Mean   :0.6488   Mean   :0.3211   Mean   :130.3
##  3rd Qu.:140.0   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:203.0
##  Max.   :148.0   Max.   :1.0000   Max.   :1.0000   Max.   :285.0
##   DEATH_EVENT
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3211
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

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

3. Limpieza de los datos.

No tienen valores NA'.

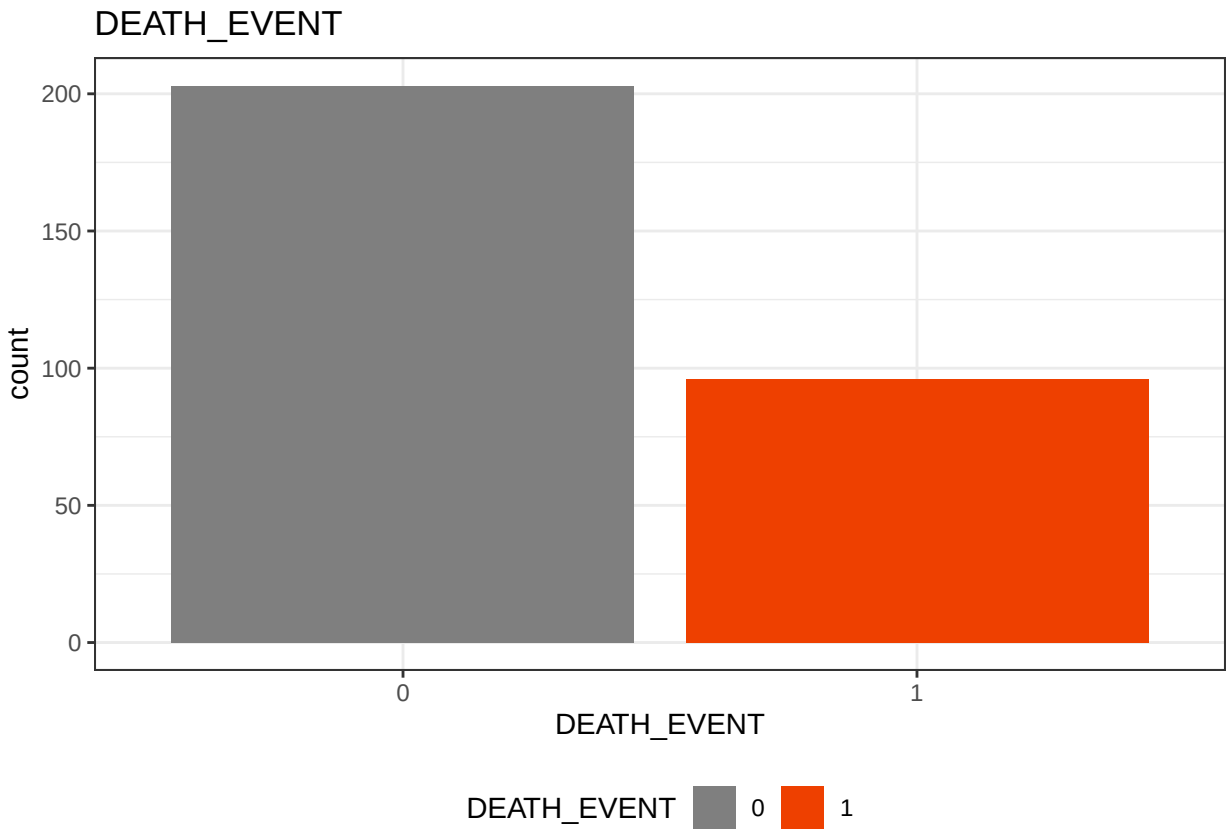
4. Análisis de los datos

Variable dependiente (DEATH_EVENT): Análisis del balanceo del data set.

```
#Factorizamos la varible

hf$DEATH_EVENT <- as.factor(hf$DEATH_EVENT)

ggplot(data = hf, aes(x = DEATH_EVENT, y = ..count.., fill = DEATH_EVENT)) +
  geom_bar() +
  scale_fill_manual(values = c("gray50", "orangered2")) +
  labs(title = "DEATH_EVENT") +
  theme_bw() +
  theme(legend.position = "bottom")
```



- Tabla de frecuencias (#):

```
table(hf$DEATH_EVENT)
```

```
##
##   0   1
## 203  96
```

- Tabla de frecuencias (%):

```
prop.table(table(hf$DEATH_EVENT)) %>% round(digits = 2)
```

```
##
##    0    1
## 0.68 0.32
```

Para que un modelo predictivo sea útil, debe de tener un porcentaje de acierto superior a lo esperado por azar a un determinado nivel basal. En problemas de clasificación, el nivel basal es el que se obtiene si se asignan todas las observaciones a la clase mayoritaria (la moda). Por tanto ha de superar el 32%. Este es el porcentaje mínimo que hay que intentar superar con los modelos predictivos. (Siendo estrictos, este porcentaje tendrá que ser recalculado únicamente con el conjunto de entrenamiento).

Variables independientes:

Variables cuantitativas: Las variables age, creatinine_phosphokinase, ejection_fraction, platelets, serum_creatinine, serum_sodium, time son variables cuantitativas.

age

Edad del paciente objeto de estudio.

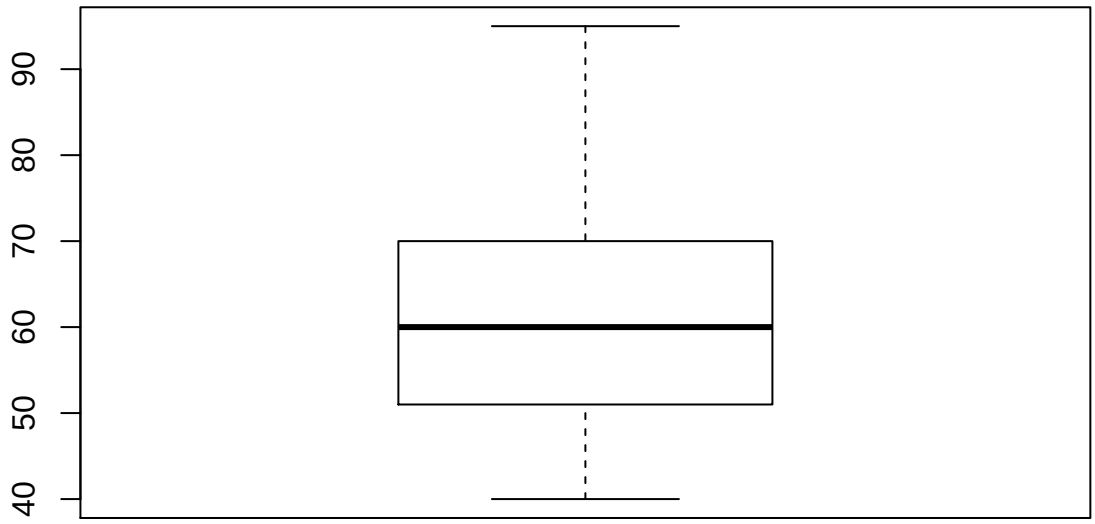
Estadisticos de la variable:

```
summary(hf$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40.00   51.00   60.00   60.83   70.00   95.00
```

Boxplot

```
boxplot(hf$age)
```

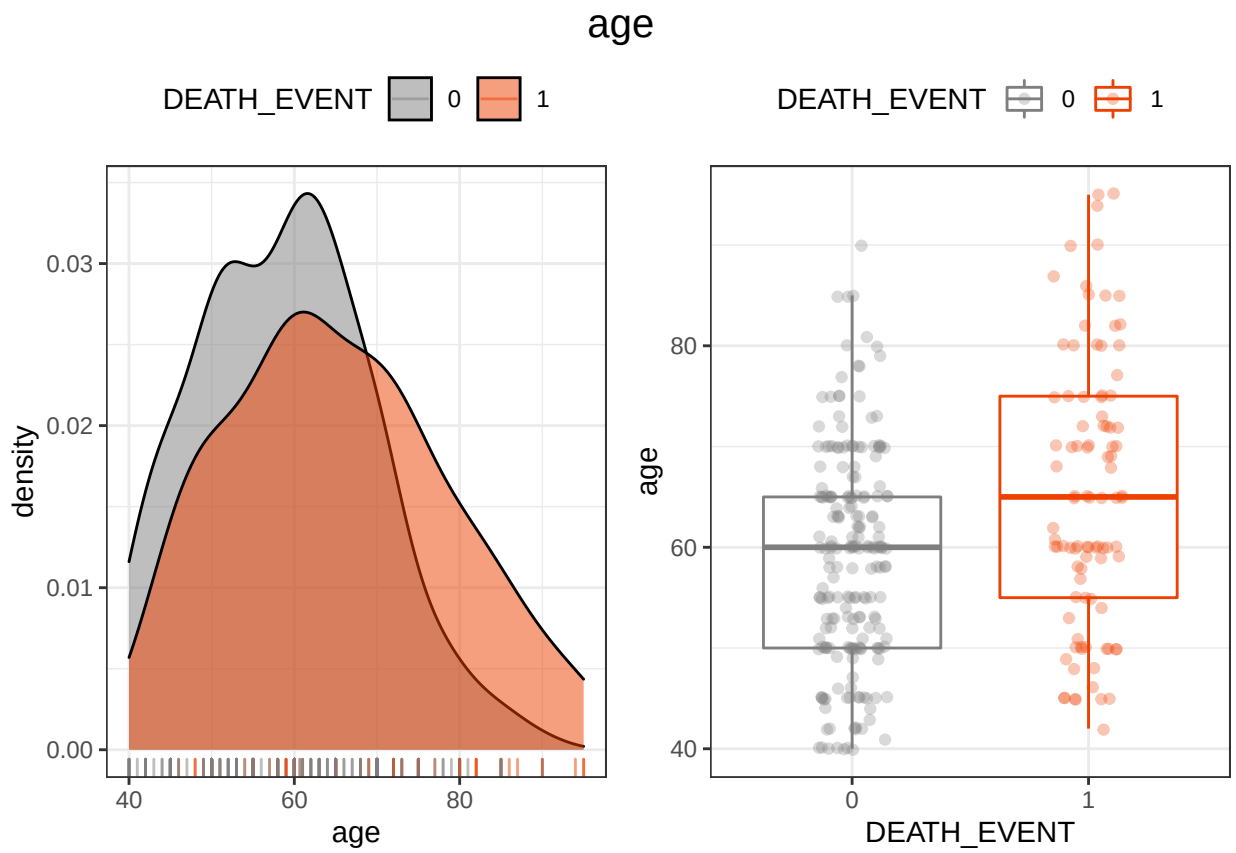


No se observan valores atípicos (outliers).

Análisis de la variable frente a la variable dependiente:

```
p1 <- ggplot(data = hf, aes(x = age, fill = DEATH_EVENT)) +
      geom_density(alpha = 0.5) +
      scale_fill_manual(values = c("gray50", "orangered2")) +
      geom_rug(aes(color = DEATH_EVENT), alpha = 0.5) +
      scale_color_manual(values = c("gray50", "orangered2")) +
```

```
      theme_bw()
p2 <- ggplot(data = hf, aes(x = DEATH_EVENT, y = age, color = DEATH_EVENT)) +
      geom_boxplot(outlier.shape = NA) +
      geom_jitter(alpha = 0.3, width = 0.15) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
final_plot <- ggarrange(p1, p2, legend = "top")
final_plot <- annotate_figure(final_plot, top = text_grob("age", size = 15))
final_plot
```



Estadísticos según la variable dependiente:

```
# Estadísticos del precio del billete de los supervivientes y fallecidos
hf %>% filter(!is.na(age)) %>% group_by(DEATH_EVENT) %>%
          summarise(media = mean(age),
                    mediana = median(age),
                    min = min(age),
                    max = max(age))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   DEATH_EVENT media mediana   min   max
##   <fct>       <dbl>   <dbl> <dbl> <dbl>
## 1 0            58.8      60    40    90
## 2 1            65.2      65    42    95
```

Tras el análisis se observa como aumenta la probabilidad de fallecimiento en función de la edad.

creatinine_phosphokinase

Nivel de la encima CPK en sangre (mcg/L)

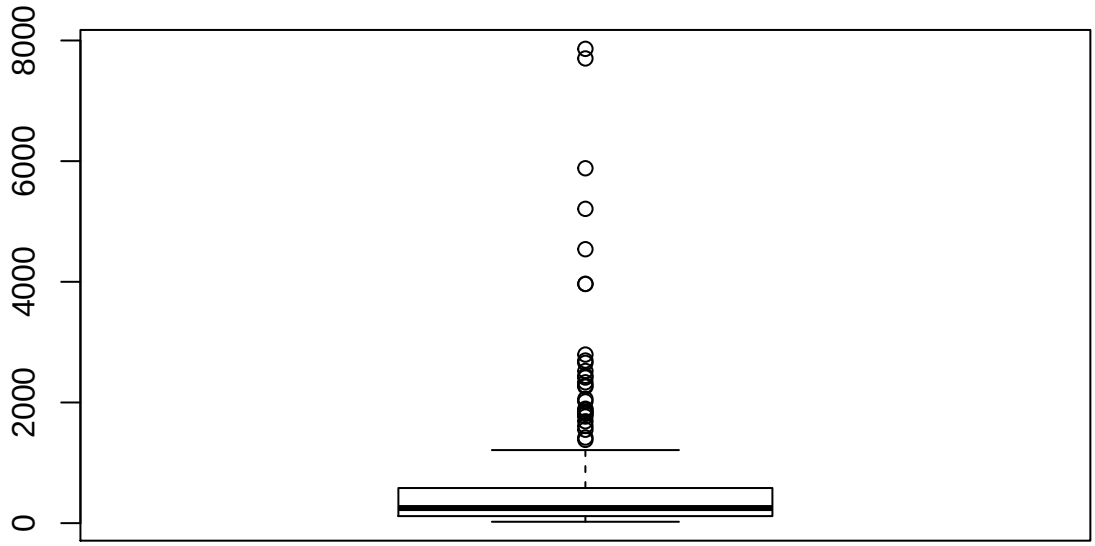
Estadisticos de la variable:

```
summary(hf$creatinine_phosphokinase)
```

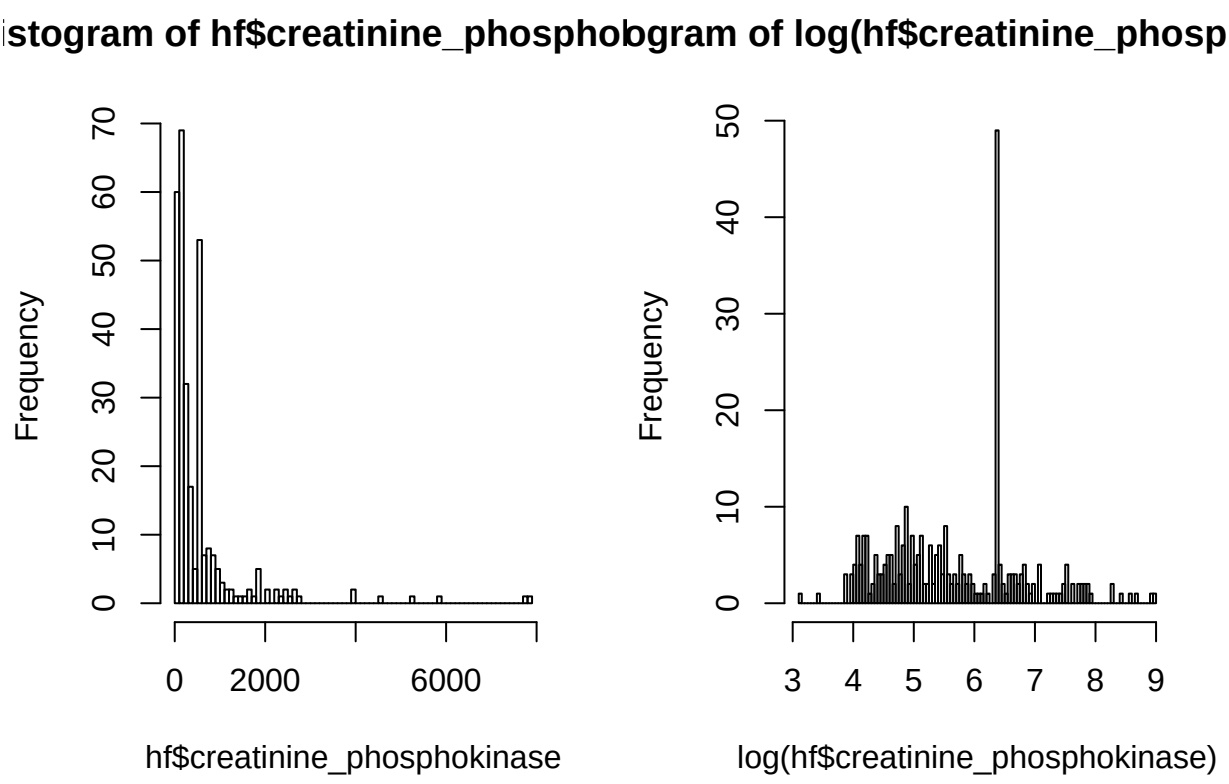
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    23.0   116.5   250.0   581.8   582.0  7861.0
```

Boxplot:

```
boxplot(hf$creatinine_phosphokinase)
```



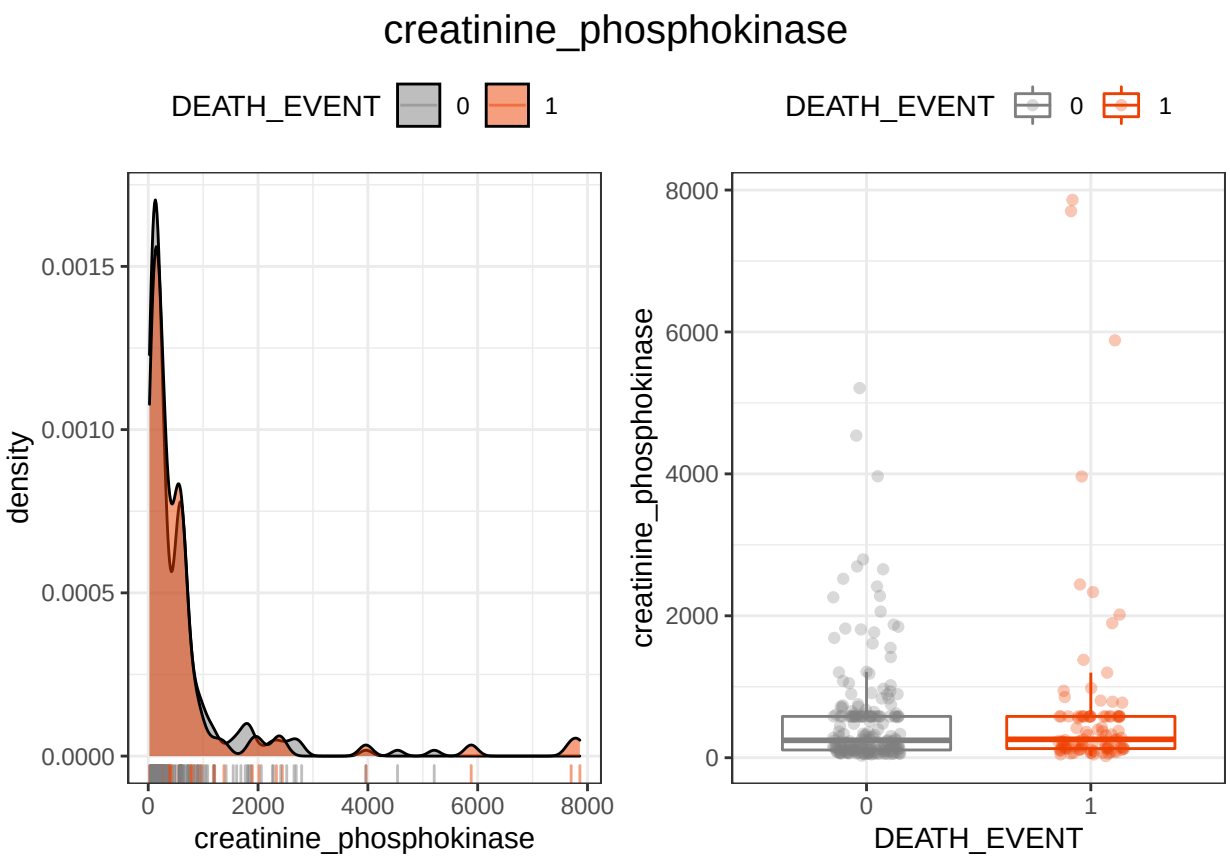

```
par(mfrow=c(1,2))
hist(hf$creatinine_phosphokinase, breaks = 100)
hist(log(hf$creatinine_phosphokinase), breaks = 100)
```



Se observan valores atípicos (outliers).

Análisis de la variable frente a la variable dependiente:

```
p1 <- ggplot(data = hf, aes(x = creatinine_phosphokinase, fill = DEATH_EVENT)) +
      geom_density(alpha = 0.5) +
      scale_fill_manual(values = c("gray50", "orangered2")) +
      geom_rug(aes(color = DEATH_EVENT), alpha = 0.5) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
p2 <- ggplot(data = hf, aes(x = DEATH_EVENT, y = creatinine_phosphokinase, color = DEATH_EVENT)) +
      geom_boxplot(outlier.shape = NA) +
      geom_jitter(alpha = 0.3, width = 0.15) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
final_plot <- ggarrange(p1, p2, legend = "top")
final_plot <- annotate_figure(final_plot, top = text_grob("creatinine_phosphokinase", size = 15))
final_plot
```



Estadísticos según la variable dependiente:

```
# Estadísticos del precio del billete de los supervivientes y fallecidos
hf %>% filter(!is.na(creatinine_phosphokinase)) %>% group_by(DEATH_EVENT) %>%
          summarise(media = mean(creatinine_phosphokinase),
                    mediana = median(creatinine_phosphokinase),
                    min = min(creatinine_phosphokinase),
                    max = max(creatinine_phosphokinase))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 5
##   DEATH_EVENT media mediana   min   max
##   <fct>       <dbl>   <dbl> <int> <int>
## 1 0            540.     245    30  5209
## 2 1            670.     259    23  7861
```

ejection_fraction

Porcentaje de sangre que sale del corazón en cada contracción.

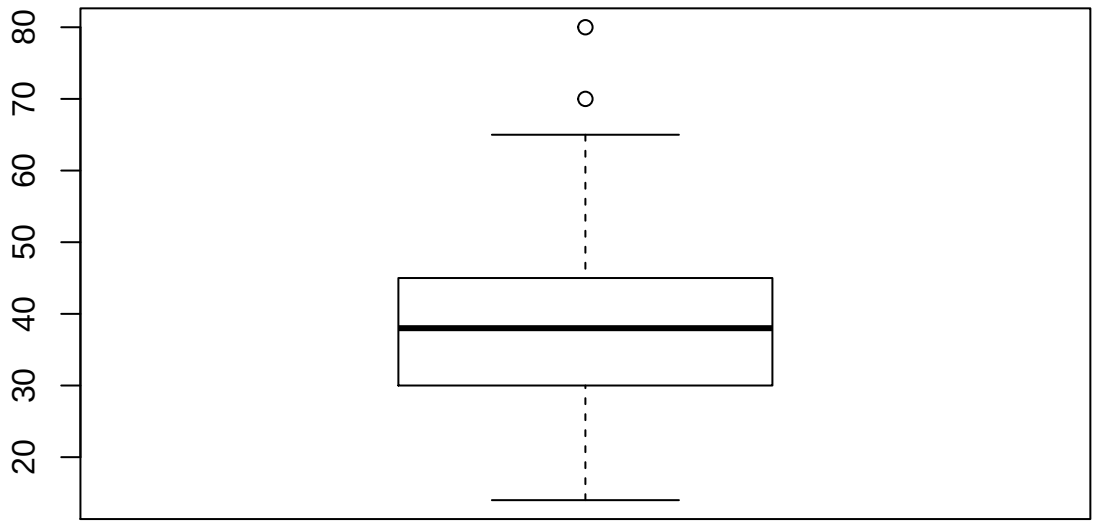
Estadisticos de la variable:

```
summary(hf$ejection_fraction)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.00   30.00   38.00   38.08   45.00   80.00
```

Boxplot:

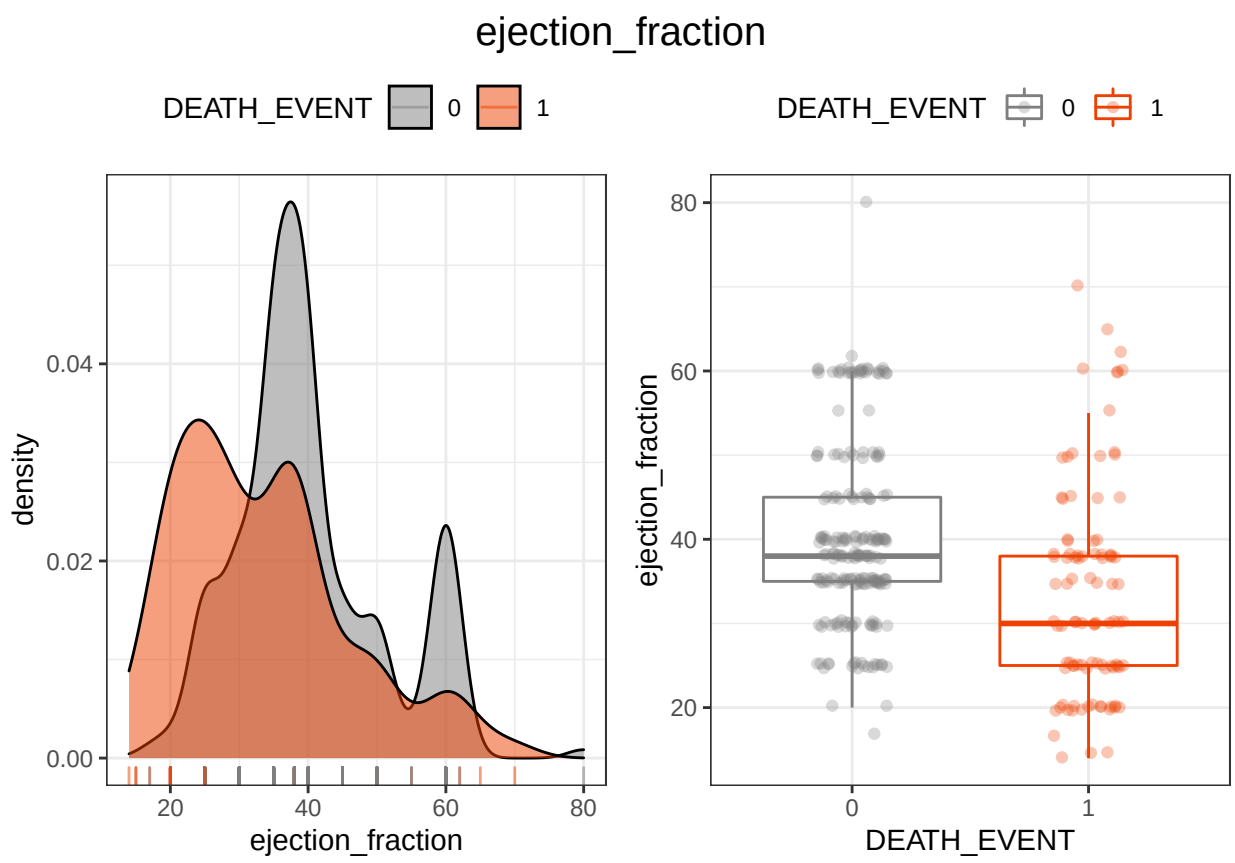
```
boxplot(hf$ejection_fraction)
```



No se observan valores atípicos (outliers).

Análisis de la variable frente a la variable dependiente:

```
p1 <- ggplot(data = hf, aes(x = ejection_fraction, fill = DEATH_EVENT)) +
      geom_density(alpha = 0.5) +
      scale_fill_manual(values = c("gray50", "orangered2")) +
      geom_rug(aes(color = DEATH_EVENT), alpha = 0.5) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
p2 <- ggplot(data = hf, aes(x = DEATH_EVENT, y = ejection_fraction, color = DEATH_EVENT)) +
      geom_boxplot(outlier.shape = NA) +
      geom_jitter(alpha = 0.3, width = 0.15) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
final_plot <- ggarrange(p1, p2, legend = "top")
final_plot <- annotate_figure(final_plot, top = text_grob("ejection_fraction", size = 15))
final_plot
```



Estadísticos según la variable dependiente:

```
# Estadísticos del precio del billete de los supervivientes y fallecidos
hf %>% filter(!is.na(ejection_fraction)) %>% group_by(DEATH_EVENT) %>%
          summarise(media = mean(ejection_fraction),
                    mediana = median(ejection_fraction),
                    min = min(ejection_fraction),
                    max = max(ejection_fraction))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 5
##   DEATH_EVENT media mediana   min   max
##   <fct>       <dbl>   <dbl> <int> <int>
## 1 0            40.3      38    17    80
## 2 1            33.5      30    14    70
```

platelets

Plaquetas en la sangre (kiloplatelets/mL).

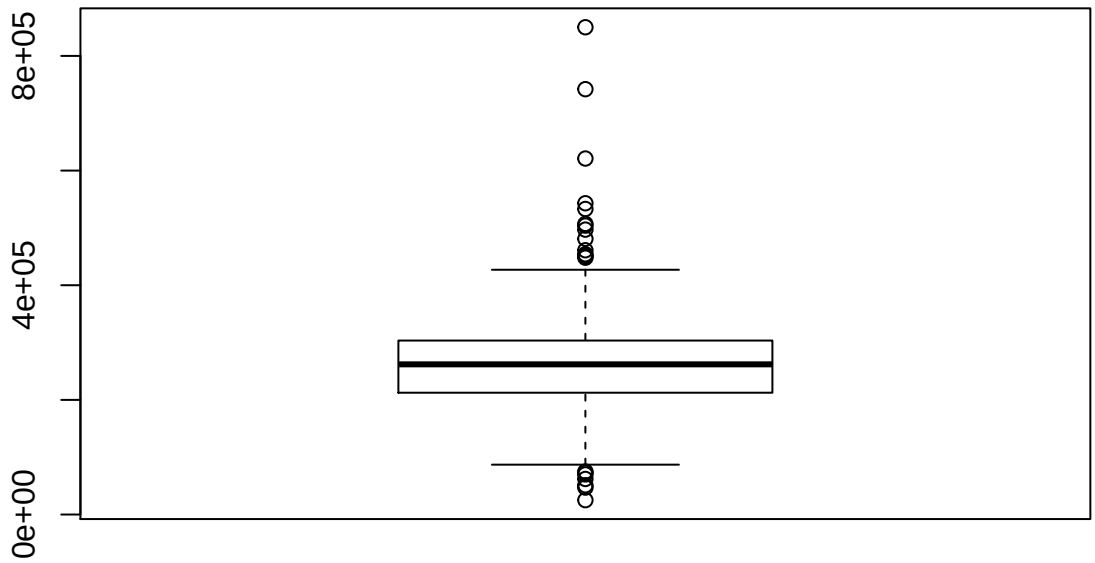
Estadisticos de la variable:

```
summary(hf$platelets)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25100  212500  262000  263358  303500  850000
```

Boxplot:

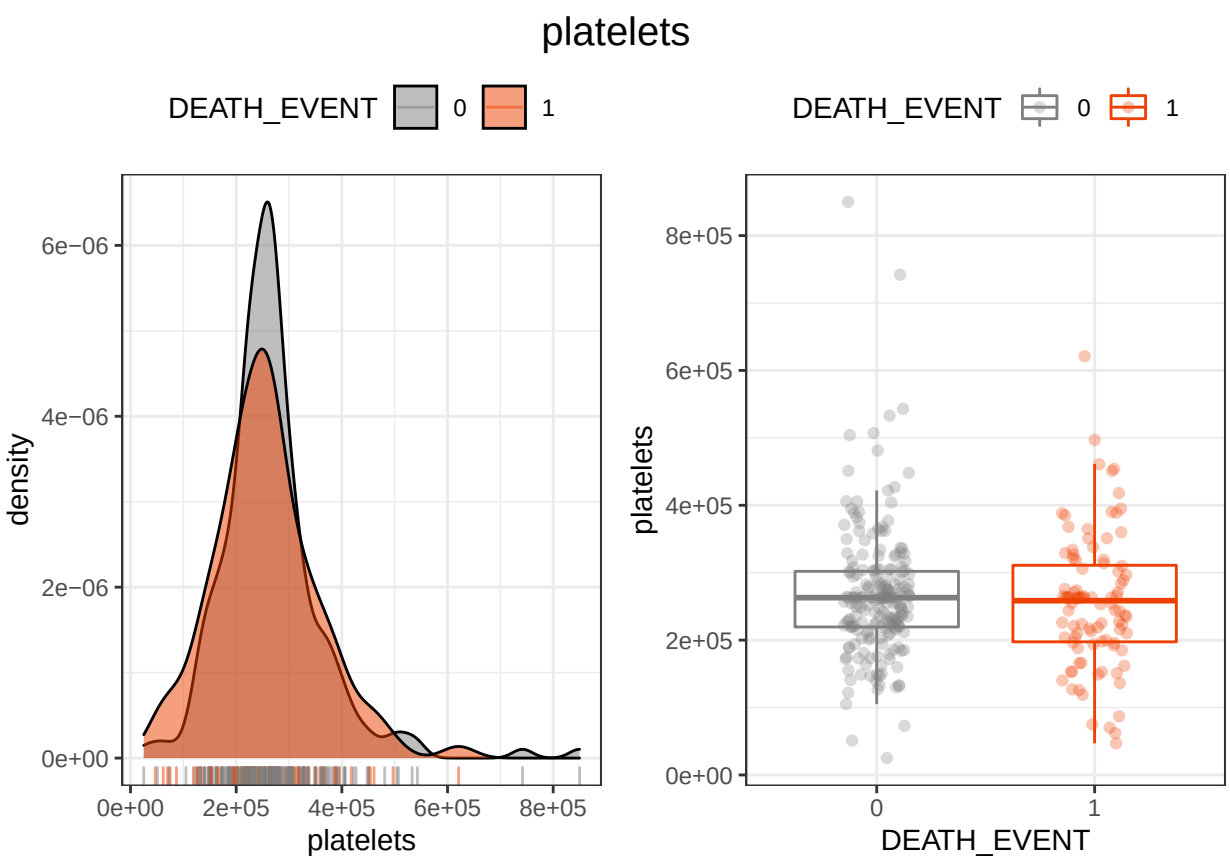
```
boxplot(hf$platelets)
```



Se observan valores atípicos (outliers).

Análisis de la variable frente a la variable dependiente:

```
p1 <- ggplot(data = hf, aes(x = platelets, fill = DEATH_EVENT)) +
      geom_density(alpha = 0.5) +
      scale_fill_manual(values = c("gray50", "orangered2")) +
      geom_rug(aes(color = DEATH_EVENT), alpha = 0.5) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
p2 <- ggplot(data = hf, aes(x = DEATH_EVENT, y = platelets, color = DEATH_EVENT)) +
      geom_boxplot(outlier.shape = NA) +
      geom_jitter(alpha = 0.3, width = 0.15) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
final_plot <- ggarrange(p1, p2, legend = "top")
final_plot <- annotate_figure(final_plot, top = text_grob("platelets", size = 15))
final_plot
```



Estadísticos según la variable dependiente:

```
# Estadísticos del precio del billete de los supervivientes y fallecidos
hf %>% filter(!is.na(platelets)) %>% group_by(DEATH_EVENT) %>%
          summarise(media = mean(platelets),
                    mediana = median(platelets),
                    min = min(platelets),
                    max = max(platelets))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 5
##   DEATH_EVENT   media mediana   min    max
##   <fct>         <dbl>   <dbl> <dbl>  <dbl>
## 1 0           266657.  263000 25100 850000
## 2 1           256381.  258500 47000 621000
```

serum_creatinine

Nivel de creatinina sérica en sangre (mg / dL).

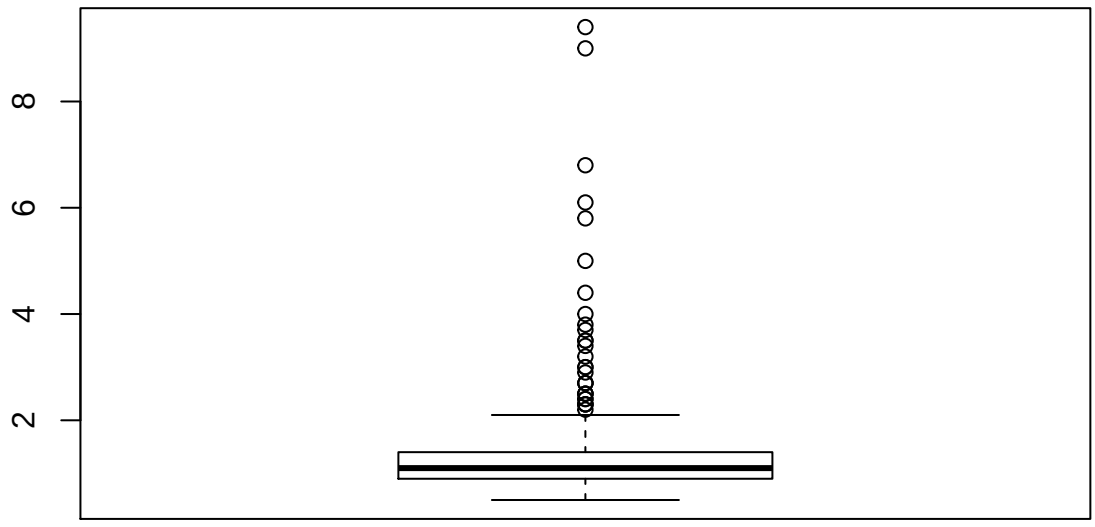
Estadisticos de la variable:

```
summary(hf$serum_creatinine)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   0.900   1.100   1.394   1.400   9.400
```

Boxplot:

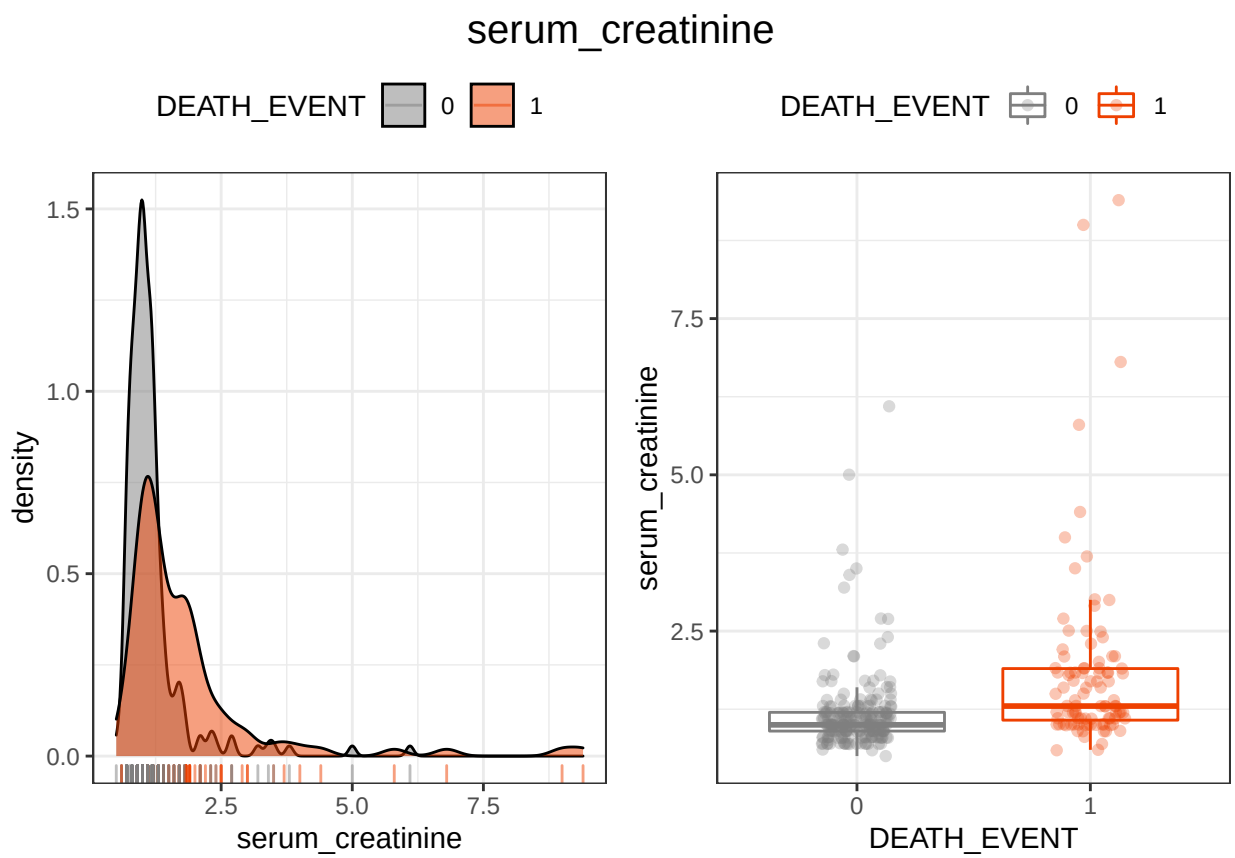
```
boxplot(hf$serum_creatinine)
```



Se observan valores atípicos (outliers).

Análisis de la variable frente a la variable dependiente:

```
p1 <- ggplot(data = hf, aes(x = serum_creatinine, fill = DEATH_EVENT)) +
      geom_density(alpha = 0.5) +
      scale_fill_manual(values = c("gray50", "orangered2")) +
      geom_rug(aes(color = DEATH_EVENT), alpha = 0.5) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
p2 <- ggplot(data = hf, aes(x = DEATH_EVENT, y = serum_creatinine, color = DEATH_EVENT)) +
      geom_boxplot(outlier.shape = NA) +
      geom_jitter(alpha = 0.3, width = 0.15) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
final_plot <- ggarrange(p1, p2, legend = "top")
final_plot <- annotate_figure(final_plot, top = text_grob("serum_creatinine", size = 15))
final_plot
```



Estadísticos según la variable dependiente:

```
# Estadísticos del precio del billete de los supervivientes y fallecidos
hf %>% filter(!is.na(serum_creatinine)) %>% group_by(DEATH_EVENT) %>%
          summarise(media = mean(serum_creatinine),
                    mediana = median(serum_creatinine),
                    min = min(serum_creatinine),
                    max = max(serum_creatinine))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   DEATH_EVENT media mediana   min   max
##   <fct>       <dbl>   <dbl> <dbl> <dbl>
## 1 0            1.18     1     0.5   6.1
## 2 1            1.84     1.3   0.6   9.4
```

serum_sodium

Nivel de sodio sérico en sangre (mEq / L).

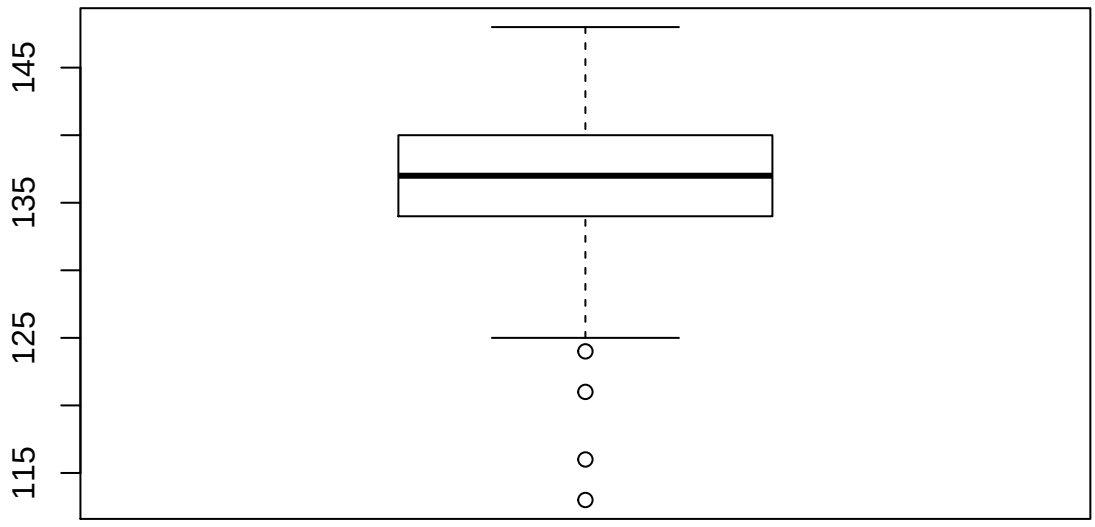
Estadisticos de la variable:

```
summary(hf$serum_sodium)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   113.0   134.0   137.0   136.6   140.0   148.0
```

Boxplot:

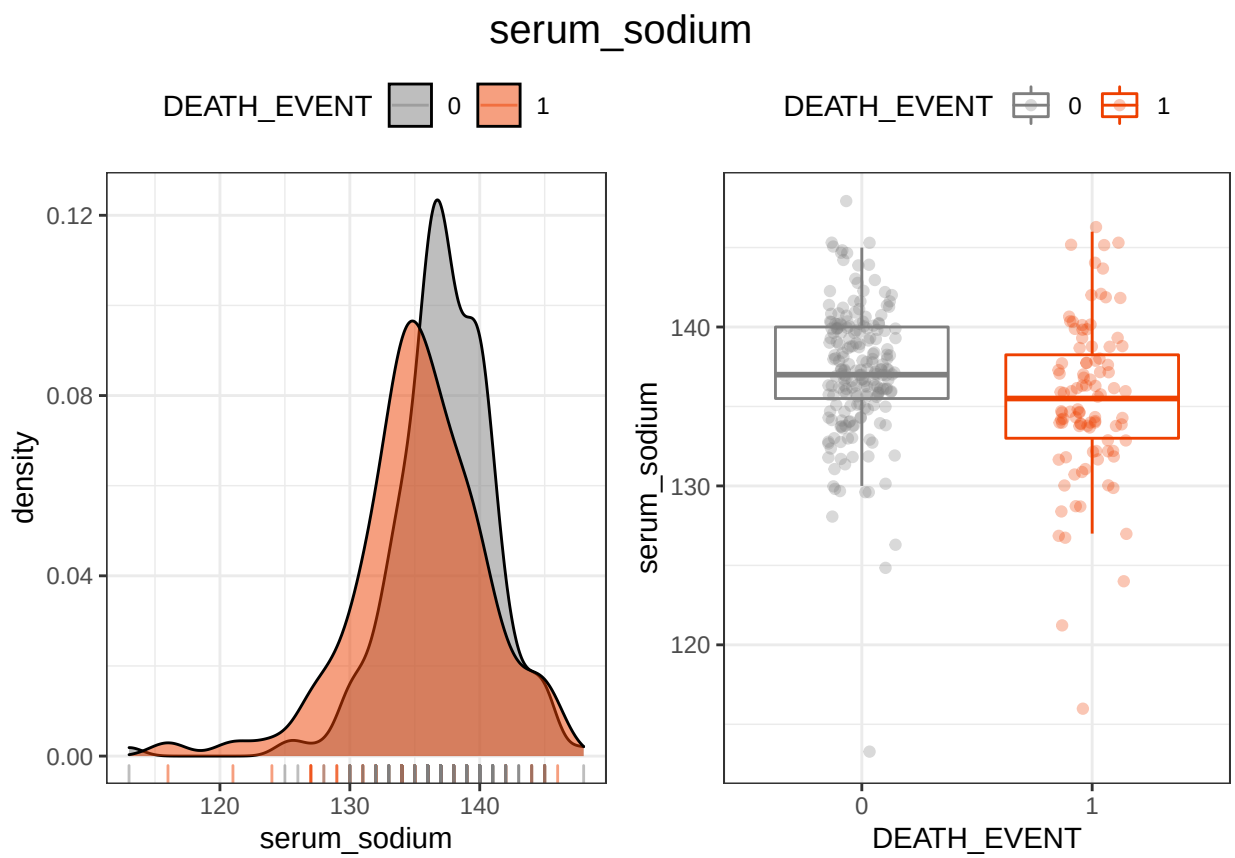
```
boxplot(hf$serum_sodium)
```



Se observan valores atípicos (outliers).

Análisis de la variable frente a la variable dependiente:

```
p1 <- ggplot(data = hf, aes(x = serum_sodium, fill = DEATH_EVENT)) +
      geom_density(alpha = 0.5) +
      scale_fill_manual(values = c("gray50", "orangered2")) +
      geom_rug(aes(color = DEATH_EVENT), alpha = 0.5) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
p2 <- ggplot(data = hf, aes(x = DEATH_EVENT, y = serum_sodium, color = DEATH_EVENT)) +
      geom_boxplot(outlier.shape = NA) +
      geom_jitter(alpha = 0.3, width = 0.15) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
final_plot <- ggarrange(p1, p2, legend = "top")
final_plot <- annotate_figure(final_plot, top = text_grob("serum_sodium", size = 15))
final_plot
```



Estadísticos según la variable dependiente:

```
# Estadísticos del precio del billete de los supervivientes y fallecidos
hf %>% filter(!is.na(serum_sodium)) %>% group_by(DEATH_EVENT) %>%
          summarise(media = mean(serum_sodium),
                    mediana = median(serum_sodium),
                    min = min(serum_sodium),
                    max = max(serum_sodium))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   DEATH_EVENT media mediana   min   max
##   <fct>       <dbl>   <dbl> <int> <int>
## 1 0            137.    137    113   148
## 2 1            135.    136.   116   146
```

time

Período de seguimiento (días).

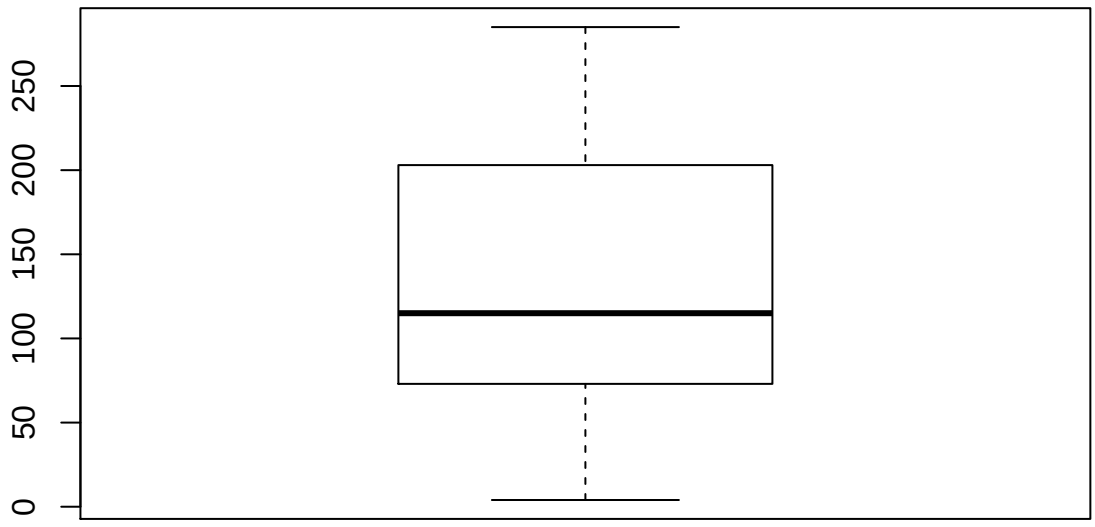
Estadisticos de la variable:

```
summary(hf$time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0    73.0   115.0   130.3   203.0   285.0
```

Boxplot:

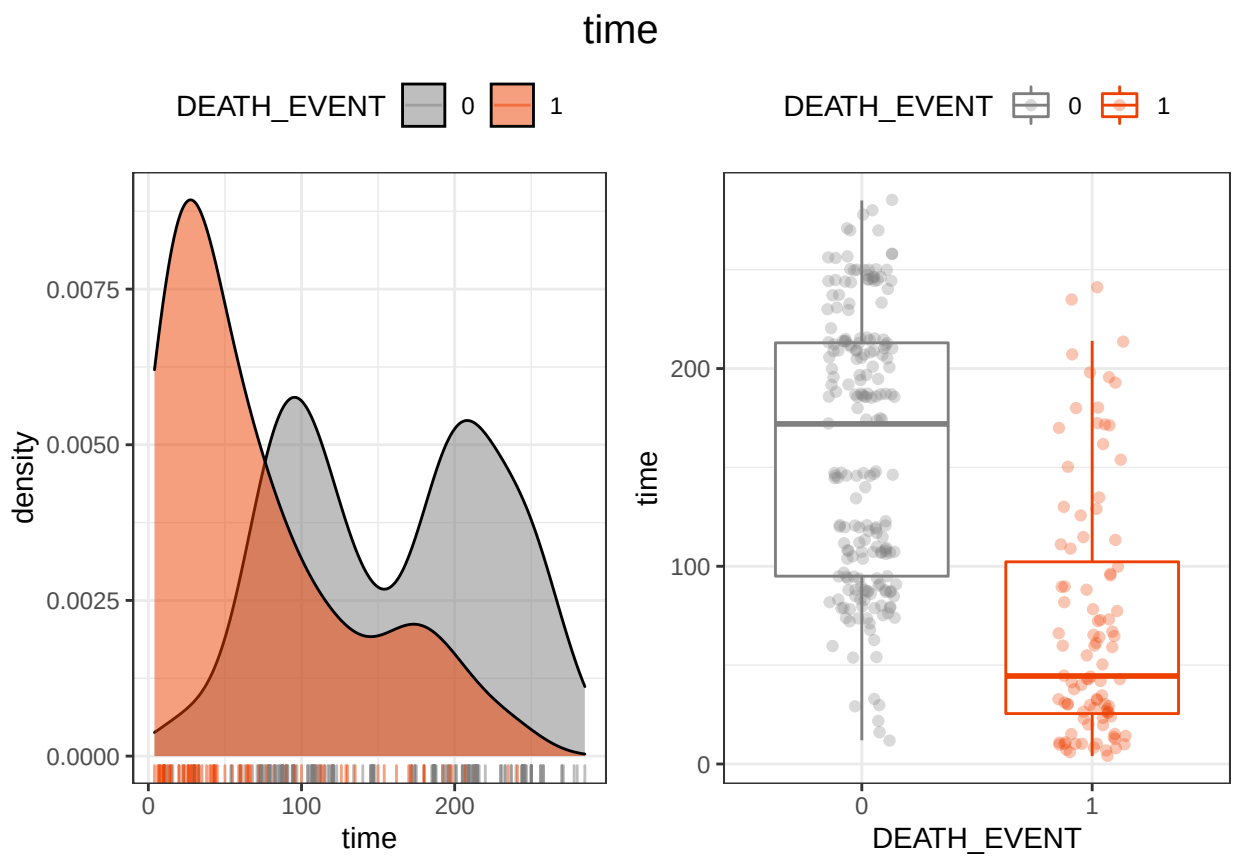
```
boxplot(hf$time)
```



Se observan valores atípicos (outliers).

Análisis de la variable frente a la variable dependiente:

```
p1 <- ggplot(data = hf, aes(x = time, fill = DEATH_EVENT)) +
      geom_density(alpha = 0.5) +
      scale_fill_manual(values = c("gray50", "orangered2")) +
      geom_rug(aes(color = DEATH_EVENT), alpha = 0.5) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
p2 <- ggplot(data = hf, aes(x = DEATH_EVENT, y = time, color = DEATH_EVENT)) +
      geom_boxplot(outlier.shape = NA) +
      geom_jitter(alpha = 0.3, width = 0.15) +
      scale_color_manual(values = c("gray50", "orangered2")) +
      theme_bw()
final_plot <- ggarrange(p1, p2, legend = "top")
final_plot <- annotate_figure(final_plot, top = text_grob("time", size = 15))
final_plot
```



Estadísticos según la variable dependiente:

```
# Estadísticos del precio del billete de los supervivientes y fallecidos
hf %>% filter(!is.na(time)) %>% group_by(DEATH_EVENT) %>%
          summarise(media = mean(time),
                    mediana = median(time),
                    min = min(time),
                    max = max(time))
```

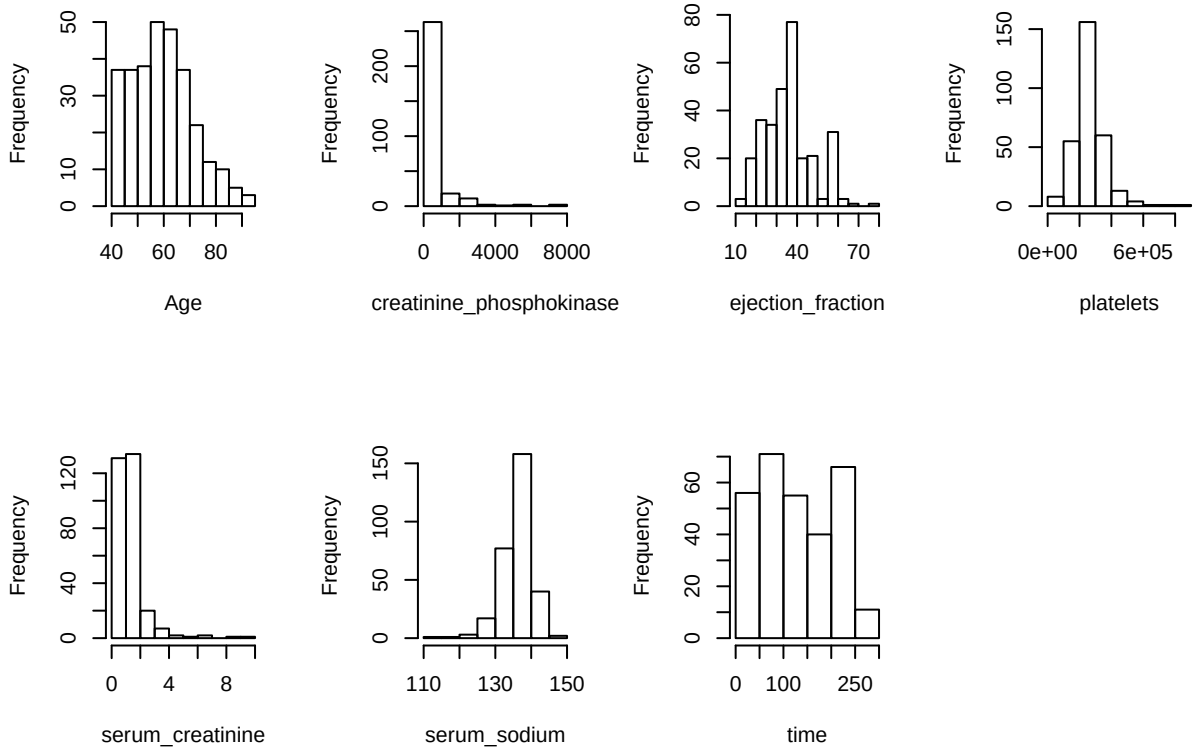
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   DEATH_EVENT media mediana   min   max
##   <fct>       <dbl>   <dbl> <int> <int>
## 1 0           158.    172      12   285
## 2 1            70.9    44.5     4   241
```

Variables cualitativas: Las variables anaemia, diabetes, high_blood_pressure, sex, smoking, DEATH_EVENT son variables cualitativas y dicotomicas (valores 1 o 0).

```
par(mfrow=c(2,4))
hist(hf$age, main = "", xlab = "Age")
hist((hf$creatinine_phosphokinase), main = "", xlab = "creatinine_phosphokinase")
hist(hf$ejection_fraction, main = "", xlab = "ejection_fraction")
hist(hf$platelets, main = "", xlab = "platelets")
```

```r
hist((hf$serum_creatinine), main = "", xlab = "serum_creatinine")
hist(hf$serum_sodium, main = "", xlab = "serum_sodium")
hist(hf$time, main = "", xlab = "time")
```



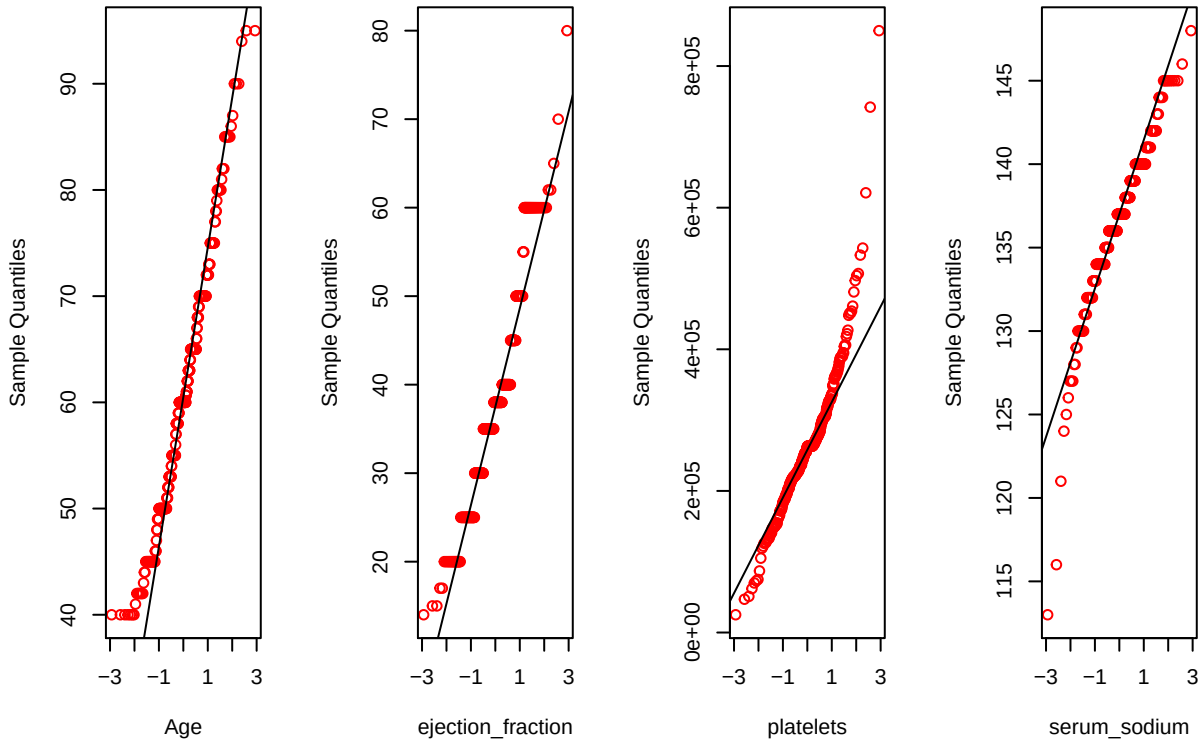
Age, ejection_fraction, platelets, serum_sodium son variables más o menos normales.
Creatinine_phosphokinase, serum creatinine no están normalizadas (con claramente logaritmicas) time es una variable que representa el tiempo hasta que el paciente o bien se muere o bien se le da de alta, por lo que desde el punto de vista de diagnóstico no debería ser utilizado (si para posteriores análisis)

Comprobamos la normalidad de las variables Age, ejection_fraction, platelets, serum_sodium

```r
par(mfrow=c(1,4))

#boxplot(hf$age, main = "", xlab = "Age")
qqnorm(hf$age, main = "", xlab = "Age", col="red")
qqline(hf$age, main = "", xlab = "Age", col="black")
#boxplot(hf$creatinine_phosphokinase, main = "", xlab = "creatinine_phosphokinase")
#qqnorm(hf$creatinine_phosphokinase, main = "", xlab = "creatinine_phosphokinase", col="red")
#boxplot(hf$ejection_fraction, main = "", xlab = "ejection_fraction")
qqnorm(hf$ejection_fraction, main = "", xlab = "ejection_fraction", col="red")
qqline(hf$ejection_fraction, main = "", xlab = "ejection_fraction", col="black")
#boxplot(hf$platelets, main = "", xlab = "platelets")
qqnorm(hf$platelets, main = "", xlab = "platelets", col="red")
qqline(hf$platelets, main = "", xlab = "platelets", col="black")
#boxplot(hf$serum_creatinine, main = "", xlab = "serum_creatinine")
#qqnorm(hf$serum_creatinine, main = "", xlab = "serum_creatinine", col="red")
```

```
#boxplot(hf$serum_sodium, main = "", xlab = "serum_sodium")
qqnorm(hf$serum_sodium, main = "", xlab = "serum_sodium", col="red")
qqline(hf$serum_sodium, main = "", xlab = "", col="black")
```

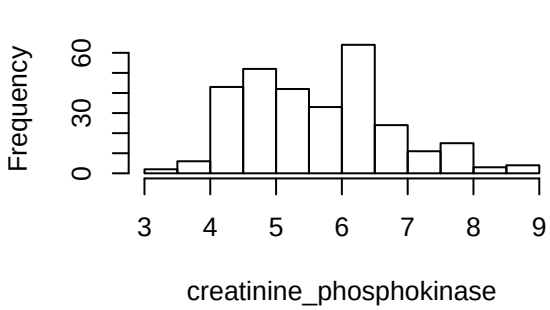


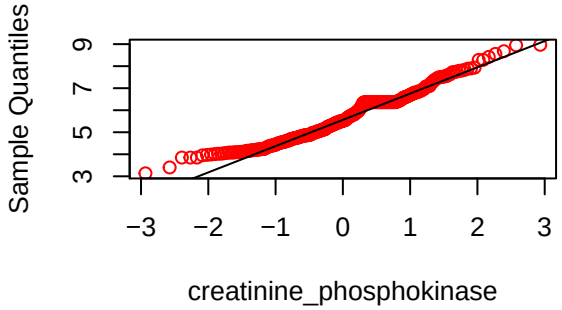

```
#boxplot(hf$time, main = "", xlab = "time")
#qqnorm(hf$time, main = "", xlab = "time", col="red")
```

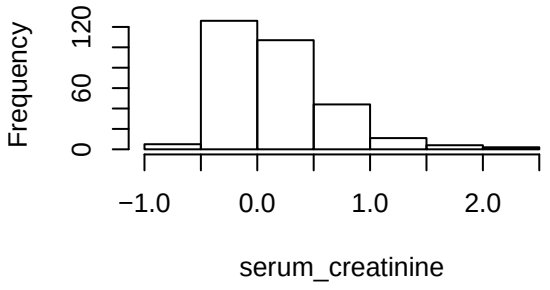
Si cambiamos ahora los valores de creatinine_phosphokinase y serum_creatinine por sus valores logarítmicos, tenemos que las variables mejoran en su normalidad:

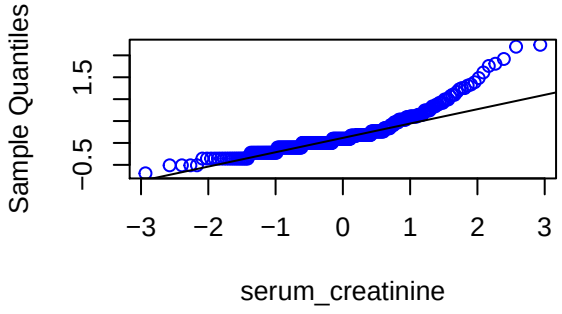
```
par(mfrow=c(2,2))

hist(log(hf$creatinine_phosphokinase), main = "", xlab = "creatinine_phosphokinase")
qqnorm(log(hf$creatinine_phosphokinase), main = "", xlab = "creatinine_phosphokinase", col="red")
qqline(log(hf$creatinine_phosphokinase), main = "", xlab = "", col="black")
hist((log(hf$serum_creatinine)), main = "", xlab = "serum_creatinine")
qqnorm(log(hf$serum_creatinine), main = "", xlab = "serum_creatinine", col="blue")
qqline(log(hf$serum_creatinine), main = "", xlab = "serum_creatinine", col="black")
```



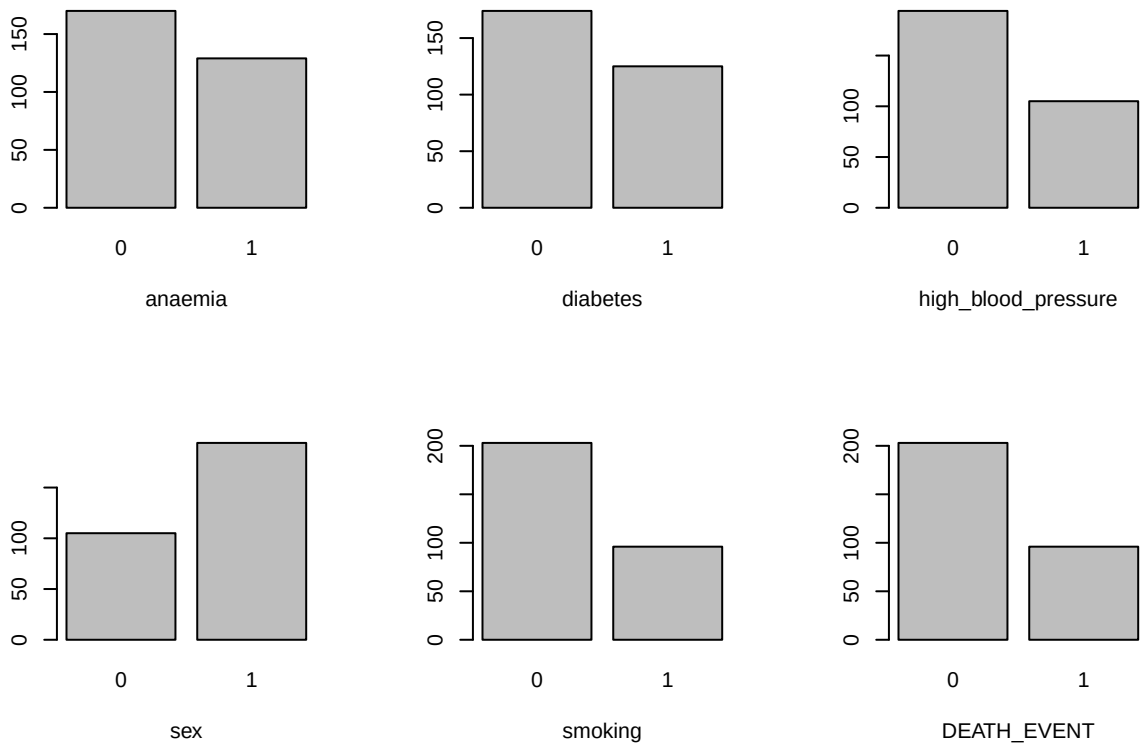






Las variables cualitativas las estudiamos con barplot: anaemia, diabetes, high_blood_pressure, sex, smoking, DEATH_EVENT

```
par(mfrow=c(2,3))
barplot(table(hf$anaemia), xlab="anaemia")
barplot(table(hf$diabetes), xlab="diabetes")
barplot(table(hf$high_blood_pressure), xlab="high_blood_pressure")
barplot(table(hf$sex), xlab="sex")
barplot(table(hf$smoking), xlab="smoking")
barplot(table(hf$DEATH_EVENT), xlab="DEATH_EVENT")
```




```
SM_SEX <- table(hf$smoking, hf$sex) # en el eje y muestra smoking

# Relación smoking fente a sex
SM_SEX
```

```
##
##       0   1
##   0 101 102
##   1   4  92
```

```
# Relación smoking fente a sex (valores relativos)
prop.table(SM_SEX)
```

```
##
##              0          1
##   0 0.33779264 0.34113712
##   1 0.01337793 0.30769231
```

Lo que nos dice esta tabla es que hay sólo cuatro mujeres fumadoras. Una población muy pequeña.

```
#Creación de subsets de trabajo para estudio de variables cualitativas anaemia, diabetes, high_blood_pr


hf_sm <- hf[hf$smoking==1 ,]
```

```r
hf_nosm<- hf[hf$smoking==0 ,]

hf_m<- hf[hf$sex==1 ,]
hf_w<- hf[hf$sex==0 ,]

hf_a<- hf[hf$anaemia==1 ,]
hf_noa<- hf[hf$anaemia==0 ,]

hf_d<- hf[hf$diabetes==1 ,]
hf_nod<- hf[hf$diabetes==0 ,]

hf_hp<- hf[hf$high_blood_pressure==1 ,]
hf_nohp<- hf[hf$high_blood_pressure==0 ,]
```

```r
#cálculo de medias de tiempo en fallacer en función de variables diatomicas

sm <-hf_sm$time[hf_sm$DEATH_EVENT==1]
nosm <-hf_nosm$time[hf_nosm$DEATH_EVENT==1]

m <-hf_m$time[hf_m$DEATH_EVENT==1]
w <-hf_w$time[hf_w$DEATH_EVENT==1]

a <-hf_a$time[hf_a$DEATH_EVENT==1]
noa <-hf_noa$time[hf_noa$DEATH_EVENT==1]

d <-hf_d$time[hf_d$DEATH_EVENT==1]
nod <-hf_nod$time[hf_nod$DEATH_EVENT==1]

hp <-hf_hp$time[hf_hp$DEATH_EVENT==1]
nohp <-hf_nohp$time[hf_nohp$DEATH_EVENT==1]

# medias por fumar
c(mean(sm), mean(nosm))
```

```
## [1] 61.03333 75.36364
```

```r
#medias por sexo
c(mean(m), mean(w))
```

```
## [1] 69.19355 73.97059
```

```r
#medias por anemia
c(mean(a), mean(noa))
```

```
## [1] 63.56522 77.62000
```

```r
#medias por diabetes
c(mean(d), mean(nod))
```

```
## [1] 69.02500 72.21429
```

```r
#medias por hipertensión
c(mean(hp), mean(nohp))
```

```
## [1] 57.10256 80.31579
```

```r
#media total
mean(hf$time[hf$DEATH_EVENT==1])
```

```
## [1] 70.88542
```

Podemos suponer que el tiempo medio que tarda en fallecer una persona sí depende de las variables cualitativas, pero es necesario realizar un contraste estadístico.

Haremos un contraste estadístico para cada variable cualitativa para comprobar si la media es diferente en cada caso. Para ello en primer lugar sabemos que la media de una función con más de 30 muestras es normal por el teorema del límite central.

Las varianzas de las poblaciones son desconocidas pero sólo necesitamos saber si son iguales. Para ello realizamos un test de contraste de hipótesis de homocedasticidad. La hipótesis nula representa que las varianzas son iguales y la alternativa que las varianzas son diferentes.

Utilizamos la función var.test de R

https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/var.test

```r
# analisis de varianzas test de homocedasticidad


vsm<-var.test(sm,nosm, conf.level=0.95)
vm<-var.test(m, w, conf.level=0.95)
va<-var.test(a,noa, conf.level=0.95)
vd<-var.test(d,nod, conf.level=0.95)
vhp<-var.test(hp,nohp, conf.level=0.95)

c(vsm[["p.value"]], vm[["p.value"]],va[["p.value"]],vd[["p.value"]],vhp[["p.value"]])
```

```
## [1] 0.3815126 0.5297808 0.8155406 0.4591452 0.1357044
```

Dado que ningún p-value es menor que 0.05 no se puede rechazar la hipotesis nula en ninguno de los casos y por lo tanto las varianzas son iguales en todos los casos.

Para realizar los contrastes estadísticos de las medias en cada caso debemos, ado que es una muestra de una población utilizar la función de T Student con varianzas desconocidas pero iguales.

Utilizamos la función de R t.test.

https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/t.test

```r
#Contraste de hipótesis. las medias iguales en todos los casos. Con un 95% de nivel de confianza

test_sm<-t.test(sm,nosm, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
test_m<-t.test(m, w, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
test_a<-t.test(a,noa, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
test_d<-t.test(d,nod, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
test_hp<-t.test(hp,nohp, alternative="two.sided", var.equal=TRUE, conf.level=0.95)

c(test_sm[["p.value"]], test_m[["p.value"]],test_a[["p.value"]],test_d[["p.value"]],test_hp[["p.value"]]
```

```
## [1] 0.29924796 0.72173640 0.27235618 0.80641308 0.07315528
```

En ninguno de los casos el p-value es menor que el nivel de significación (0.05 por trabajar con un nivel de confianza del 95%). No pueden rechazarse ninguna de las hipótesis nulas y por lo tanto las medias de tiempo no son diferentes en ningún caso.

Veamos ahora los porcentajes de fallecidos para cada variable dicotomica

```
# Estudio de porcentajes respecto a DEATH_EVENT

po<- nrow(hf[hf$DEATH_EVENT==1,])/nrow(hf)

psm<- nrow(hf_sm[hf_sm$DEATH_EVENT==1,])/nrow(hf_sm)
pnosm<- nrow(hf_nosm[hf_nosm$DEATH_EVENT==1,])/nrow(hf_nosm)

pm<- nrow(hf_m[hf_m$DEATH_EVENT==1,])/nrow(hf_m)
pw<- nrow(hf_w[hf_w$DEATH_EVENT==1,])/nrow(hf_w)

pa <- nrow(hf_a[hf_a$DEATH_EVENT==1,])/nrow(hf_a)
pnoa <- nrow(hf_noa[hf_noa$DEATH_EVENT==1,])/nrow(hf_noa)

pd <- nrow(hf_d[hf_d$DEATH_EVENT==1,])/nrow(hf_d)
pnod <- nrow(hf_nod[hf_nod$DEATH_EVENT==1,])/nrow(hf_nod)

php <- nrow(hf_hp[hf_hp$DEATH_EVENT==1,])/nrow(hf_hp)
pnohp <- nrow(hf_nohp[hf_nohp$DEATH_EVENT==1,])/nrow(hf_nohp)

#porcentajes por fumar
c(po,psm,pnosm)
```

```
## [1] 0.3210702 0.3125000 0.3251232
```

```
#porcentajes por sexo
c(po,pm,pw)
```

```
## [1] 0.3210702 0.3195876 0.3238095
```

```
#porcentajes por anemia
c(po,pa,pnoa)
```

```
## [1] 0.3210702 0.3565891 0.2941176
```

```
#porcentajes por diabetes
c(po,pd,pnod)
```

```
## [1] 0.3210702 0.3200000 0.3218391
```

```
#porcentajes por hipertension
c(po,php,pnohp)
```

```
## [1] 0.3210702 0.3714286 0.2938144
```

Vamos a realizar igual que en el caso anterior un contraste estadístico para cada variable en el que vamos a comparar el procentaje de fallecimientos:

- ser fumador frente a no ser fumador
- ser hombre frente a ser mujer
- tener anemia frente a no tener anemia
- tener diabetes frente a no tener diabetes
- tener hipertensión frente a no tener hipertensión

la hipotesis nula en todos lo casos corresponde a que el porcentaje de fallecimientos en el primer caso es igual al porcentaje de fallecimientos en el segundo. Es decir, que el porcentaje de fallecimentos es igual en el caso de ser fumador que no, ser hombre frente a ser mujer, etc...

La hipótesis alternativa sería que el porcentaje de fallecidos es diferente.

Se trata de contrastes bilaterales de proporciones de dos muestras.

Utilizamos la función de R prop.test

https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prop.test

```
# Contraste de hipótesis para porcentajes


# fumador(a)
xsm<- c(nrow(hf_sm)*psm, nrow(hf_nosm)*pnosm)
nsm <- c(nrow(hf_sm), nrow(hf_nosm) )

prop_sm <- prop.test(xsm, nsm, alternative = "two.sided")

#sexo
xm<- c(nrow(hf_m)*pm, nrow(hf_w)*pw)
nm <- c(nrow(hf_m), nrow(hf_w) )

prop_m <- prop.test(xm, nm, alternative = "two.sided")

#anemia
xa<- c(nrow(hf_a)*pa, nrow(hf_noa)*pnoa)
na <- c(nrow(hf_a), nrow(hf_noa) )

prop_a <- prop.test(xa, na, alternative = "two.sided")

#diabetes
xd<- c(nrow(hf_d)*pd, nrow(hf_nod)*pnod)
nd <- c(nrow(hf_d), nrow(hf_nod) )

prop_d <- prop.test(xd, nd, alternative = "two.sided")

#hipertension
xhp<- c(nrow(hf_hp)*php, nrow(hf_nohp)*pnohp)
nhp <- c(nrow(hf_hp), nrow(hf_nohp) )

prop_hp <- prop.test(xhp, nhp, alternative = "two.sided")


c(prop_sm[["p.value"]], prop_m[["p.value"]],prop_a[["p.value"]],prop_d[["p.value"]],prop_hp[["p.value"]]
```

```
## [1] 0.9317653 1.0000000 0.3073161 1.0000000 0.2141034
```

De nuevo en todos los casos, p-value es superior al nivel de significación, por lo que no puede rechazarse la hipotesis nula. Los porcentajes de fallecidos en todos y cada uno de los casos son iguales.

No hay diferencias con un nivel de confianza del 95% entre ser fumador o no o ser hombre/mujer o tener diabetes... a la hora de tener más probabilidad de fallecer tras un infarto, según la muestra de datos analizada.

MODELO LOGISTICO

Vamos a calcular un modelo logístico. Dado que a priori no sabemos qué variables intervienen más en el modelo las incorporamos todas en el modelo

```r
# modelo logístico

glm_hf1 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction + creatinine_ph
summary(glm_hf1)
```

```
##
## Call:
## glm(formula = factor(DEATH_EVENT) ~ age + serum_creatinine +
##     ejection_fraction + creatinine_phosphokinase + serum_sodium +
##     platelets + factor(sex) + factor(smoking) + factor(high_blood_pressure) +
##     factor(diabetes) + anaemia, family = binomial(link = logit),
##     data = hf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3184  -0.7692  -0.4436   0.8293   2.4880
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   4.964e+00  4.601e+00   1.079 0.280625
## age                           5.569e-02  1.313e-02   4.241 2.23e-05 ***
## serum_creatinine              6.619e-01  1.734e-01   3.817 0.000135 ***
## ejection_fraction            -7.032e-02  1.486e-02  -4.731 2.23e-06 ***
## creatinine_phosphokinase      2.905e-04  1.428e-04   2.034 0.041907 *
## serum_sodium                 -5.667e-02  3.338e-02  -1.698 0.089558 .
## platelets                    -7.094e-07  1.617e-06  -0.439 0.660857
## factor(sex)1                 -3.990e-01  3.508e-01  -1.137 0.255394
## factor(smoking)1              1.356e-01  3.486e-01   0.389 0.697300
## factor(high_blood_pressure)1  4.189e-01  3.061e-01   1.369 0.171092
## factor(diabetes)1             1.514e-01  2.974e-01   0.509 0.610644
## anaemia                       4.179e-01  3.009e-01   1.389 0.164904
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 294.28  on 287  degrees of freedom
## AIC: 318.28
##
## Number of Fisher Scoring iterations: 5
```

Interpretación modelo logístico:

El logit, es decir, ln(P(Y=1/X)/1-P(Y=1/X)) es 4.964 + 5.569e-02 * age + 6.619e-01 * serum_creatinine -7.032e-02 * ejection_fraction + 2.905e-04 * creatinine_phosphokinase ...

Las variables cuyo p-value (Pr(>|z| del estadístico de Wald, z) es mayor que el nivel de significación (0.05) no son significativas y no afectan al cálculo del logit)

Se pueden calcular los Odds Ratio de cada variable como el exp de los coeficientes con un intervalo de confianza al 95%.

```
exp(glm_hf1[["coefficients"]])
```

```
##                  (Intercept)                          age
##                  143.2080946                    1.0572709
##             serum_creatinine            ejection_fraction
##                    1.9383955                    0.9320936
##     creatinine_phosphokinase                 serum_sodium
##                    1.0002906                    0.9449089
##                    platelets                 factor(sex)1
##                    0.9999993                    0.6709813
##             factor(smoking)1 factor(high_blood_pressure)1
##                    1.1452113                    1.5203456
##            factor(diabetes)1                      anaemia
##                    1.1634708                    1.5188140
```

Cuando el odds ratio = 1 indica la no existencia de relación entre variables. Cuando el odds ratio > 1 indica que existe una relación entre variables y que incrementos en la variable independiente aumenta la probabilidad de ocurrir el evento (fallecimiento) Cuando el odds ratio < 1 indica que existe una relación negativa. Es decir, incrementos de la variable independiente disminuye la probabilidad del evento (fallecimiento)

Cuanto mayor (o menor) sea el valor del odds ratio respecto a 1 mayor será la variación de la probabilidad por cada unidad que aumente la variable, asi, para la variable age (por ejemplo), la relación es mayor que 1, y al ser variable continua, indica que por cada unidad que aumenta, el odds de DEATH_EVENT aumenta un 1.0572709. Es decir la probabilidad de fallecer dividido por la probabilidad de no fallecer es un 1.0572709 mayor.

El p-valor del estadístico de Wald (z value) nos informa si la variable es significativa por lo que podemos inferir que la variable es estadísticamente significativa.

A la hora de crear un modelo logístico tnemos que tener en cuenta, por tanto, el valor del odd ratio, el p valor y el AIC del modelo resultante.

Creamos varios modelos y determinamos el que tiene el AIC menor

```
# modelo logístico

glm_hf2 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction  , family=binomi
glm_hf3 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction + creatinine_pho
glm_hf4 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction + creatinine_pho
glm_hf5 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction + creatinine_pho
glm_hf6 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction  + serum_sodium
glm_hf7 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction + creatinine_pho
glm_hf8 <- glm(formula= factor(DEATH_EVENT)~ age + serum_creatinine + ejection_fraction + creatinine_pho

c(glm_hf2 [["aic"]], glm_hf3 [["aic"]], glm_hf4 [["aic"]], glm_hf5 [["aic"]], glm_hf6 [["aic"]], glm_hf
```

```
## [1] 313.2827 313.1982 312.7058 314.7012 313.0898 312.0522 312.8302
```

El modelo que proporciona el AIC más bajo es el que contempla las variables age, serum_creatinine, ejection_fraction, creatinine_phosphokinase, serum_sodium, high_blood_pressure y anaemia.

El modelo obtenido ya nos proporciona unos valores estimados que podemos comparar con los reales. En fitted.values se encuentra la probabilidad (como el exp(logit)/(1+exp(logit))). Para valores mayores de 0.5 estimamos que la persona fallece y para los casos en que es menor la persona sobrevive.

```
#Exactitud del modelo estimado.

pred<- glm_hf1[["fitted.values"]]
pred <- data.frame(pred)
for (i in 1:nrow(pred)) {
  if (pred$pred[i] < 0.5){
  pred$pred[i] <- 0
  }else{
  pred$pred[i] <- 1
  }
}
resul <- 0
pred2<- data.frame(pred$pred,hf$DEATH_EVENT,resul)
names (pred2) <- c("pred","real","resul")

for (i in 1:nrow(pred2)) {
  if (pred2$pred[i] ==  pred2$real[i]){
  pred2$resul[i] <- 1
  }else{
  pred2$resul[i] <- 0
  }
}
sum(pred2$resul)
```

```
## [1] 230
```

```
sum(pred2$resul)/nrow(pred2)
```

```
## [1] 0.7692308
```

```
library(C50)
```

```
## Warning: package 'C50' was built under R version 3.6.3
```

```
set.seed(725)
data_random <- hf[sample(nrow(hf)),]
Y <- as.factor(data_random[,13])
X <- data_random[,1:11]

trainX <- X[1:199,]
trainY <- Y[1:199]

testX <- X[200:299,]
```

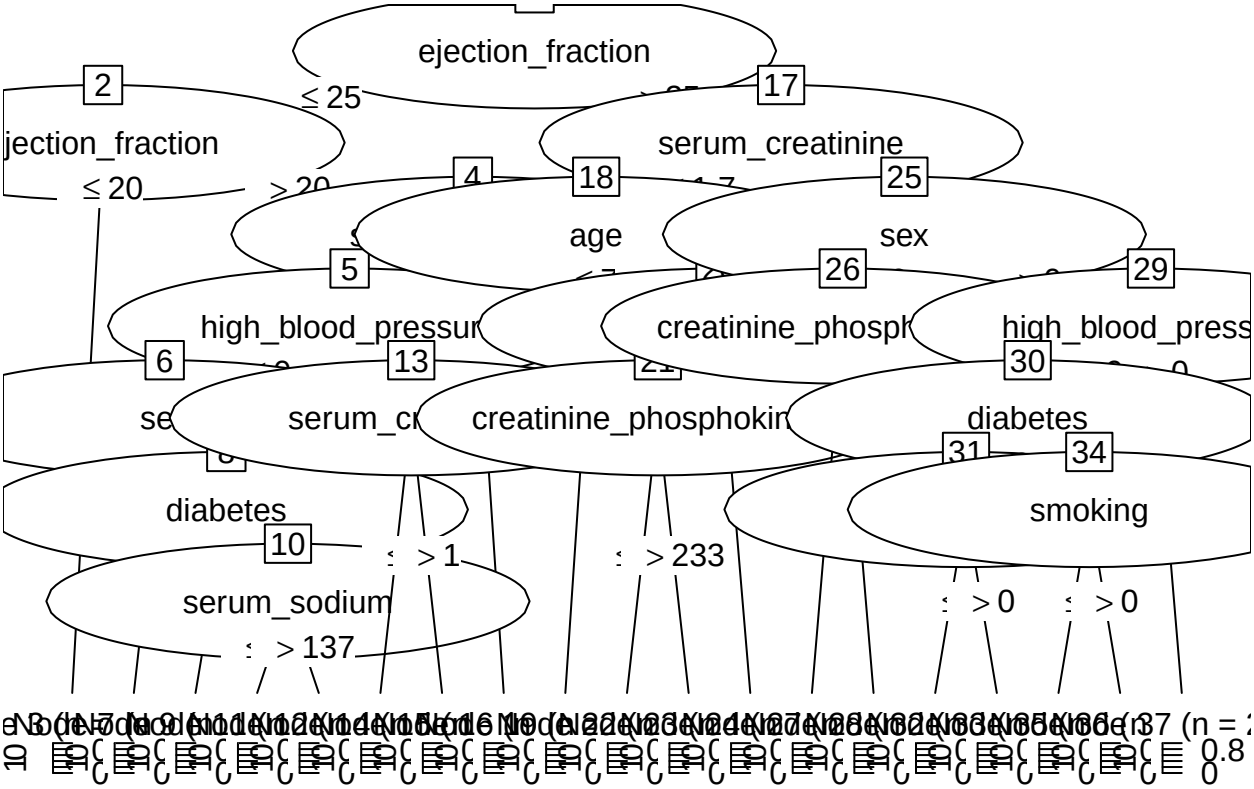
```r
testY <- Y[200:299]

hfC50 <- C50::C5.0(trainX, trainY,rules=TRUE )
summary(hfC50)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainY, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Dec 14 22:36:03 2020
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 199 cases (12 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (183/50, lift 1.1)
##  ejection_fraction > 20
##  ->  class 0  [0.724]
##
## Rule 2: (8, lift 2.8)
##  creatinine_phosphokinase > 167
##  serum_creatinine > 1.7
##  sex <= 0
##  ->  class 1  [0.900]
##
## Rule 3: (7, lift 2.8)
##  diabetes > 0
##  ejection_fraction <= 25
##  high_blood_pressure <= 0
##  serum_sodium <= 137
##  ->  class 1  [0.889]
##
## Rule 4: (6, lift 2.8)
##  diabetes <= 0
##  ejection_fraction > 25
##  serum_creatinine > 1.7
##  sex > 0
##  smoking <= 0
##  ->  class 1  [0.875]
##
## Rule 5: (5, lift 2.7)
##  ejection_fraction > 20
##  ejection_fraction <= 25
##  high_blood_pressure > 0
##  serum_creatinine <= 1
##  ->  class 1  [0.857]
##
## Rule 6: (4, lift 2.6)
##  age > 78
```

```
##  creatinine_phosphokinase <= 233
##  serum_creatinine <= 1.7
##  ->  class 1  [0.833]
##
## Rule 7: (3, lift 2.5)
##  diabetes > 0
##  serum_creatinine > 1.7
##  smoking > 0
##  ->  class 1  [0.800]
##
## Rule 8: (16/3, lift 2.5)
##  ejection_fraction <= 20
##  ->  class 1  [0.778]
##
## Default class: 0
##
##
## Evaluation on training data (199 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       8   25(12.6%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     133     3    (a): class 0
##      22    41    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% ejection_fraction
##   12.56% serum_creatinine
##    8.04% diabetes
##    7.04% sex
##    6.03% creatinine_phosphokinase
##    6.03% high_blood_pressure
##    4.52% smoking
##    3.52% serum_sodium
##    2.01% age
##
##
## Time: 0.0 secs
```

```
hfC50 <- C50::C5.0(trainX, trainY )
plot(hfC50)
```




```
predicted_model <- predict( hfC50, testX, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_model == testY) / length(predicted_
```

```
## [1] "La precisión del árbol es: 79.0000 %"
```

```
mat_conf<-table(testY,Predicted=predicted_model)
mat_conf
```

```
##      Predicted
## testY  0  1
##     0 61  6
##     1 15 18
```

```
library(descr)
```

```
## Warning: package 'descr' was built under R version 3.6.3
```

```
CrossTable(testY, predicted_model,prop.chisq = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality',
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
```

```
## |-------------------------|
##
## ==============================
##            Prediction
## Reality       0      1   Total
## ------------------------------
## 0            61      6      67
##            0.61   0.06
## ------------------------------
## 1            15     18      33
##            0.15   0.18
## ------------------------------
## Total        76     24     100
## ==============================
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.6.3
```

```
# naives bayes como modelo de clusterización (a ver qué pasa)

set.seed(725)
hfNBds <- hf[sample(nrow(hf)),]
hfNBds[,13] <- as.factor(hfNBds[,13])
hfNBds_train<- hfNBds[1:199,]
hfNBds_test <- hfNBds[200:299,]
hfNBds_train<- hfNBds[,-12]  # quito time
hfNBds_test <- hfNBds[,-12]
hfNBmodel_train <- naiveBayes( DEATH_EVENT~ ., data = hfNBds_train)
hfNBmodel_train
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.6789298 0.3210702
##
## Conditional probabilities:
##    age
## Y       [,1]     [,2]
##   0 58.76191 10.63789
##   1 65.21528 13.21456
##
##    anaemia
## Y        [,1]      [,2]
##   0 0.4088670 0.4928400
##   1 0.4791667 0.5021882
##
```

```
##    creatinine_phosphokinase
## Y       [,1]      [,2]
##   0 540.0542  753.7996
##   1 670.1979 1316.5806
##
##    diabetes
## Y        [,1]      [,2]
##   0 0.4187192 0.4945689
##   1 0.4166667 0.4955946
##
##    ejection_fraction
## Y       [,1]     [,2]
##   0 40.26601 10.85996
##   1 33.46875 12.52530
##
##    high_blood_pressure
## Y        [,1]      [,2]
##   0 0.3251232 0.4695789
##   1 0.4062500 0.4937104
##
##    platelets
## Y       [,1]     [,2]
##   0 266657.5 97531.20
##   1 256381.0 98525.68
##
##    serum_creatinine
## Y       [,1]      [,2]
##   0 1.184877 0.6540827
##   1 1.835833 1.4685615
##
##    serum_sodium
## Y       [,1]     [,2]
##   0 137.2167 3.982923
##   1 135.3750 5.001579
##
##    sex
## Y        [,1]      [,2]
##   0 0.6502463 0.4780710
##   1 0.6458333 0.4807706
##
##    smoking
## Y        [,1]      [,2]
##   0 0.3251232 0.4695789
##   1 0.3125000 0.4659456
```

```
hfNBmodel_test <- predict(hfNBmodel_train, hfNBds_test[,-12])
precision <- sum(hfNBmodel_test == hfNBds_test[,12]) / nrow (hfNBds_test)
precision
```

```
## [1] 0.7123746
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
hfRF <- randomForest( DEATH_EVENT~ ., data = hfNBds_train)
print (hfRF)
```

```
##
## Call:
##  randomForest(formula = DEATH_EVENT ~ ., data = hfNBds_train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 27.42%
## Confusion matrix:
##     0  1 class.error
## 0 172 31   0.1527094
## 1  51 45   0.5312500
```

Bilbliografia

Subirats Maté, Laila; Pérez Trenard, Diego O.; Calvo González, Mireia (2019) Introducción al ciclo de la vida de los datos. UOC *Subirats Maté, Laila; Calvo González, Mireia (2019)* Web scraping. UOC *Subirats Maté, Laila; Pérez Trenard, Diego O.; Calvo González, Mireia (2019)* Introducción a limpieza y análisis de los datos. UOC *Hernández Orallo, José; Ramirez Quintana, M José; Ferri Ramírez, Cesar (2004)* Introducción a la Minería de Datos. PEARSON. *Gironés Roig, Jordi; Casas Roma, Jordi; Minguillon Alfonso, Julia; Caichuelas Quiles, Ramon (2017)* Minería de datos: Modelos y algoritmos. UOC.

Agradecimientos data set

Cita Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). (link)