Practica 2: Data Cleaning

Jose Luis Rivas Calduch y Mariano Jiménez Barca

30/11/2020

Indice de contenidos

1.- DESCRIPCIÓN DEL DATA SET

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?

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

El dataset incorpora una variable que es el tiempo desde que el paciente es ingresado hasta que el paciente, o bien fallece, o bien es dado de alta. Podemos preguntarnos si el hecho de ser fumador o tener diabetes o el sexo condiciona el número de días hasta que un paciente fallece.

Los datos los podemos encontrar aqui: https://www.kaggle.com/andrewmvd/heart-failure-clinical-data. Los datos forman parte de un artículo publicado en el BMC Medical Informatics and Decision Making. EL artículo completo puede verse aquí: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-1023-5#Tab8. Los autores del articulo son citados en el apartado agradecimientos.

Las variables del dataset son:

- age. Edad del paciente (en años)
- anaemia. Si el paciente presenta anemia en el momento del infarto. Booleano (1/0).
- creatinine_phosphokinase. Nivel de CPK en sangre (en mcg/L).
- diabetes. Si el paciente presenta diabetes. Booleano (1/0).
- ejection_fraction. Porcentaje de la sangre que deja el corazón en cada contracción. (en tanto por ciento)
- high_blood_pressure. Si el paciente presenta hipertensión. Booleano (1/0).
- platelets. Plaquetas en sangre. (en kiloplaquetas/mL).
- serum_creatinine. Nivel de creatinina en sangre (mg/dL).
- serum_sodium. Nivel de sodio en suero en sangre (mEq/L).
- sex. Hombre o mujer. Binario (w/m).
- smoking. Si el paciente fuma o no. Booleano (1/0).
- time. periodo en días en que se hace seguimiento del paciente.
- DEATH_EVENT. Si el paciente fallece durante el periodo de seguimiento. Booleano (1/0).

2.- CARGA DEL FICHERO.

```
# Carga del fichero en la variable hf



ruta<-"../data/"
#ruta<-"C:/Users/MAJIMENE/02.UOC/Master_Data_Science/Tipologia Datos/Practica2/"
file<- "heart_failure_clinical_records_dataset.csv"

rutaFile <- paste(ruta,file,sep="")

hf <- read.table(rutaFile, header= TRUE, sep=",", dec="." )
```

Visualización de las primeras filas del data set

```
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
```

3.- LIMPIEZA DE LOS DATOS.

Vamos a identificar la existencia de valores nulos (NA´s) o vacios en el data set.

Estadísticas de los valores nulos

```
colSums(is.na(hf))
```

```
##                      age                  anaemia creatinine_phosphokinase
##                        0                        0                        0
##                 diabetes        ejection_fraction      high_blood_pressure
##                        0                        0                        0
##                platelets         serum_creatinine             serum_sodium
##                        0                        0                        0
##                      sex                  smoking                     time
##                        0                        0                        0
##              DEATH_EVENT
##                        0
```

Estadísticas de los valores vacios

```
colSums(hf=="")
```

```
##                      age                  anaemia creatinine_phosphokinase
##                        0                        0                        0
##                 diabetes        ejection_fraction      high_blood_pressure
##                        0                        0                        0
##                platelets         serum_creatinine             serum_sodium
##                        0                        0                        0
##                      sex                  smoking                     time
##                        0                        0                        0
##              DEATH_EVENT
##                        0
```

Se observa que el dataset no tienen valores NA's o vacios.

4.- ESTUDIO DESCRIPTIVO DE LOS DATOS Y TRATAMIENTO

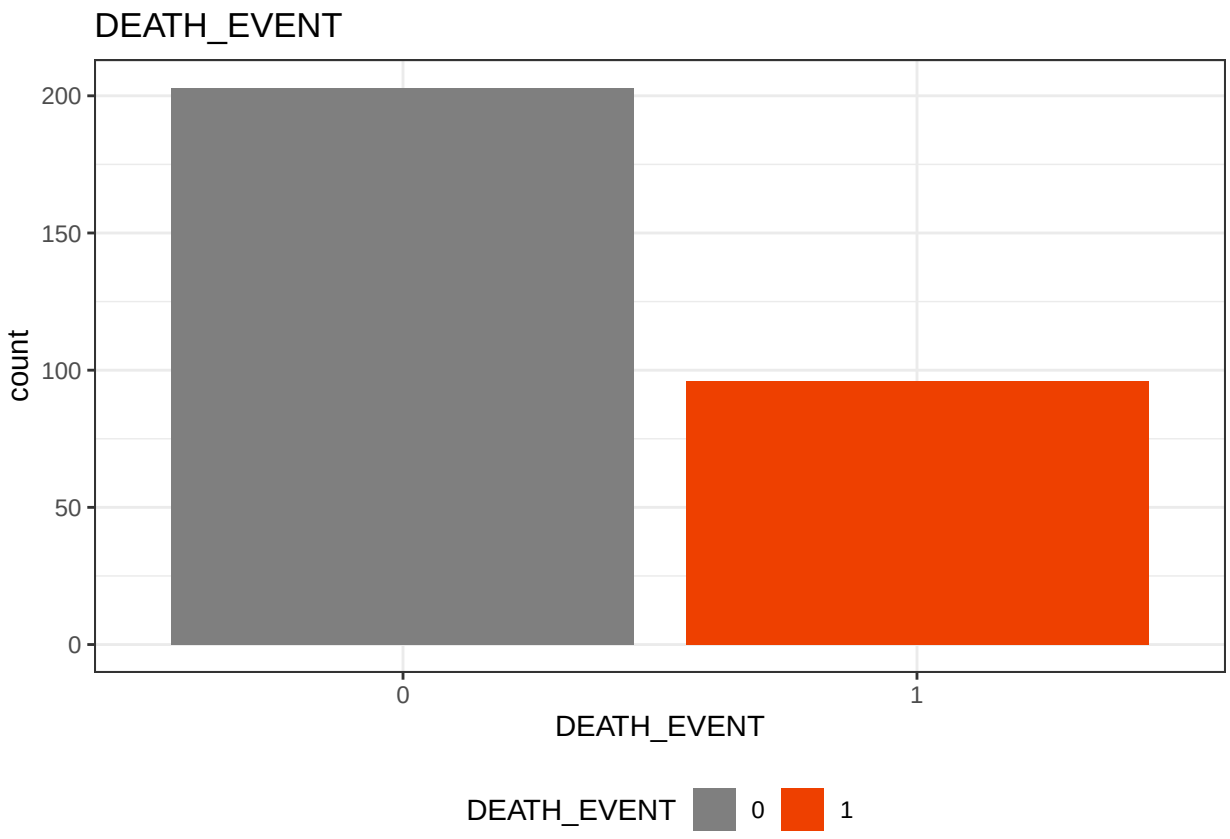
4.1. Análisis descriptivo de los datos:

Vamos a realizar una aproximación utilizando técnicas de análisis descriptivo estadístico a los datos contenidos al data set con objeto de determinar cual es su poder de discriminación frente a la variable dependiente y que implicaciones tiene entre ella de cara a desarrollar un modelo.

4.1.1 Variable dependiente (DEATH_EVENT): Nos enfrentamos a un problema a resolver de clasificación donde tenemos una variable dependiente binaria es un tipo de modelo que se utiliza para explicar fenómenos en los cuales la variable de relevancia es dicotómica o binaria, es decir, solo puede tomar dos valores.

- 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).

4.1.2 Variables independientes:

4.1.2.1 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.

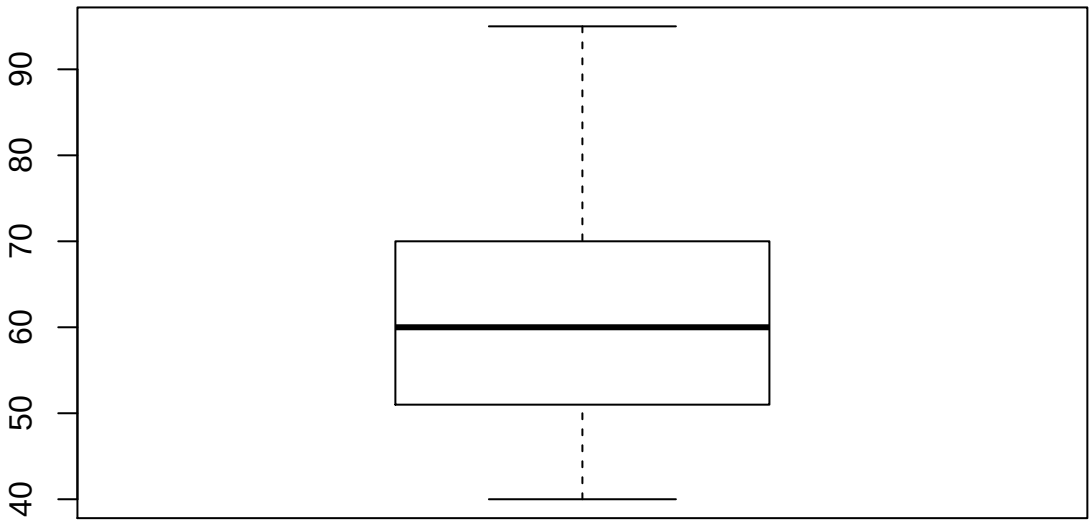
Estadísticos de la variable:

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

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

Boxplot

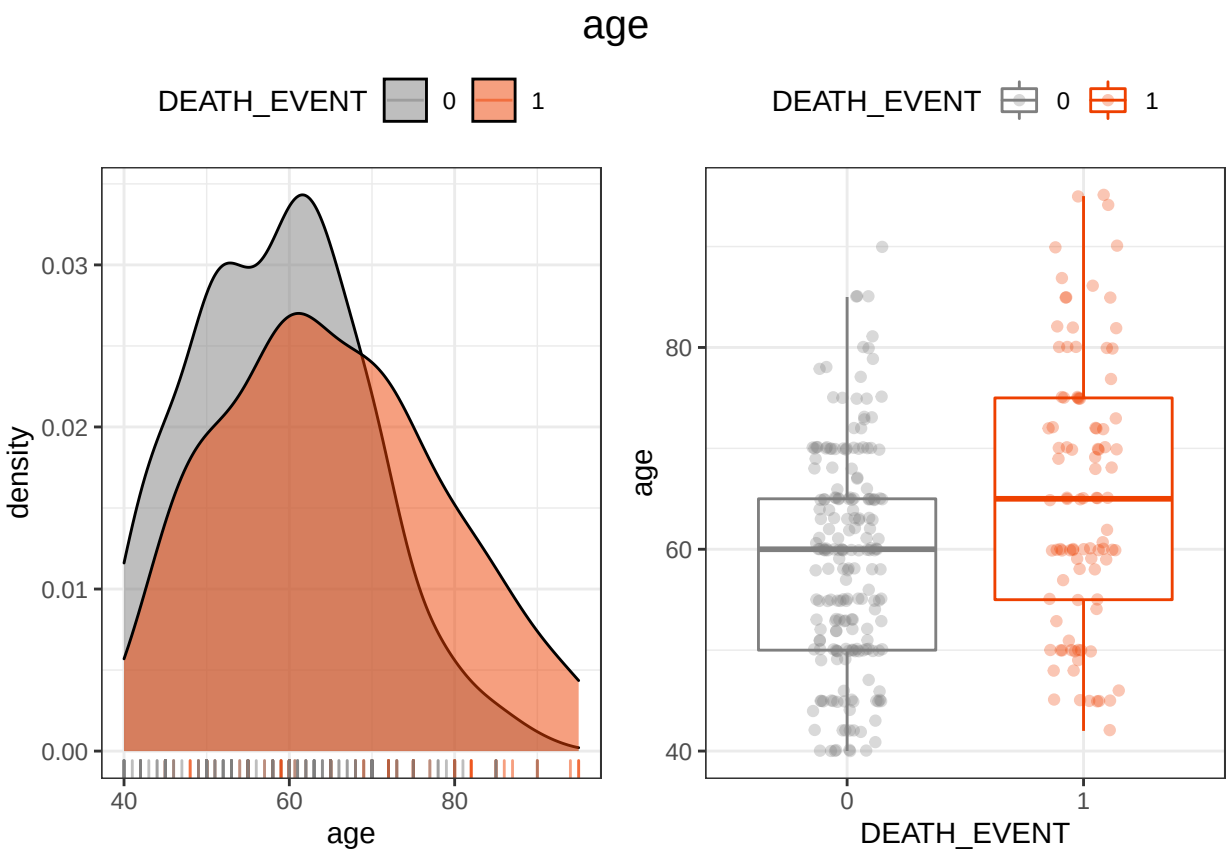
```
boxplot(hf$age)
```



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)

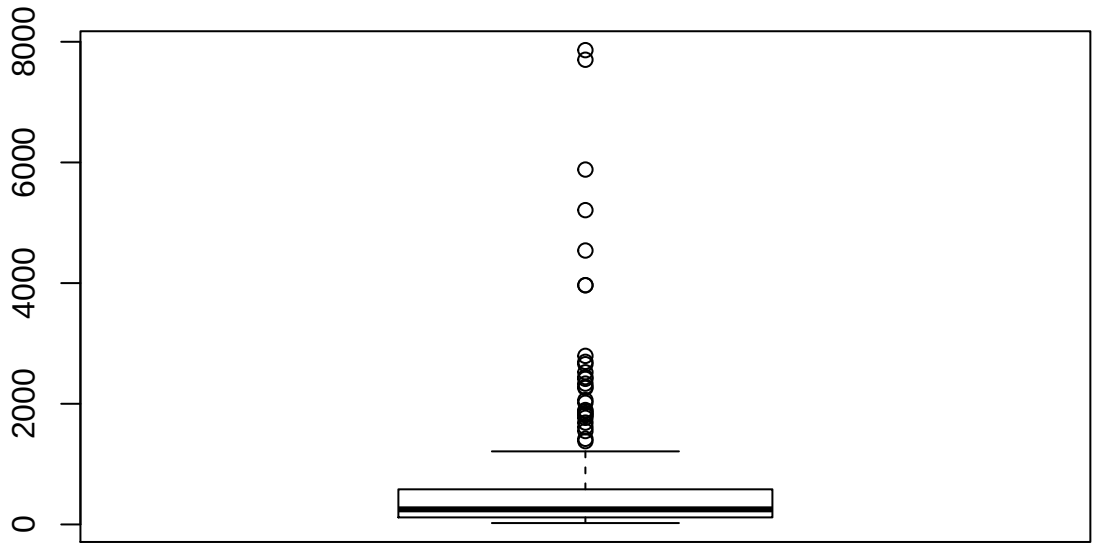
Estadísticos de la variable:

```r
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:

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



Se observan valores atípicos (outliers).

Vamos a analizar cuales son los valores atípicos de la muestra utilizando el test de Tukey, que toma como referencia la diferencia entre el primer cuartil *Q1* y el tercer cuartil *Q3*, o rango intercuartílico. En un diagrama de caja se considera un valor atípico el que se encuentra 1,5 veces esa distancia de uno de esos cuartiles (atípico leve) o a 3 veces esa distancia (atípico extremo).

```r
#Calculo de los cuartiles
qrts <- quantile(hf$creatinine_phosphokinase, probs = c(0.25, 0.75))

#Rango intercuartilico
iqr <- qrts[2]-qrts[1]

# Umbral
h_leve <- 1.5 * iqr
h_extremo <- 3 * iqr
```

```r
# Limite superior
limite_superior_leve <- qrts[2]+h_leve
limite_superior_extremo <- qrts[2]+h_extremo

# Limite inferior
limite_inferior_leve <- qrts[1]-h_leve
limite_inferior_extremo <- qrts[1]-h_extremo
```

- Limite superior leve:

```r
limite_superior_leve
```

```
##     75%
## 1280.25
```

- Limite superior extremo:

```r
limite_superior_extremo
```

```
##    75%
## 1978.5
```

- Limite inferior leve:

```r
limite_inferior_leve
```

```
##     25%
## -581.75
```

- Limite inferior extremo:

```r
limite_inferior_extremo
```

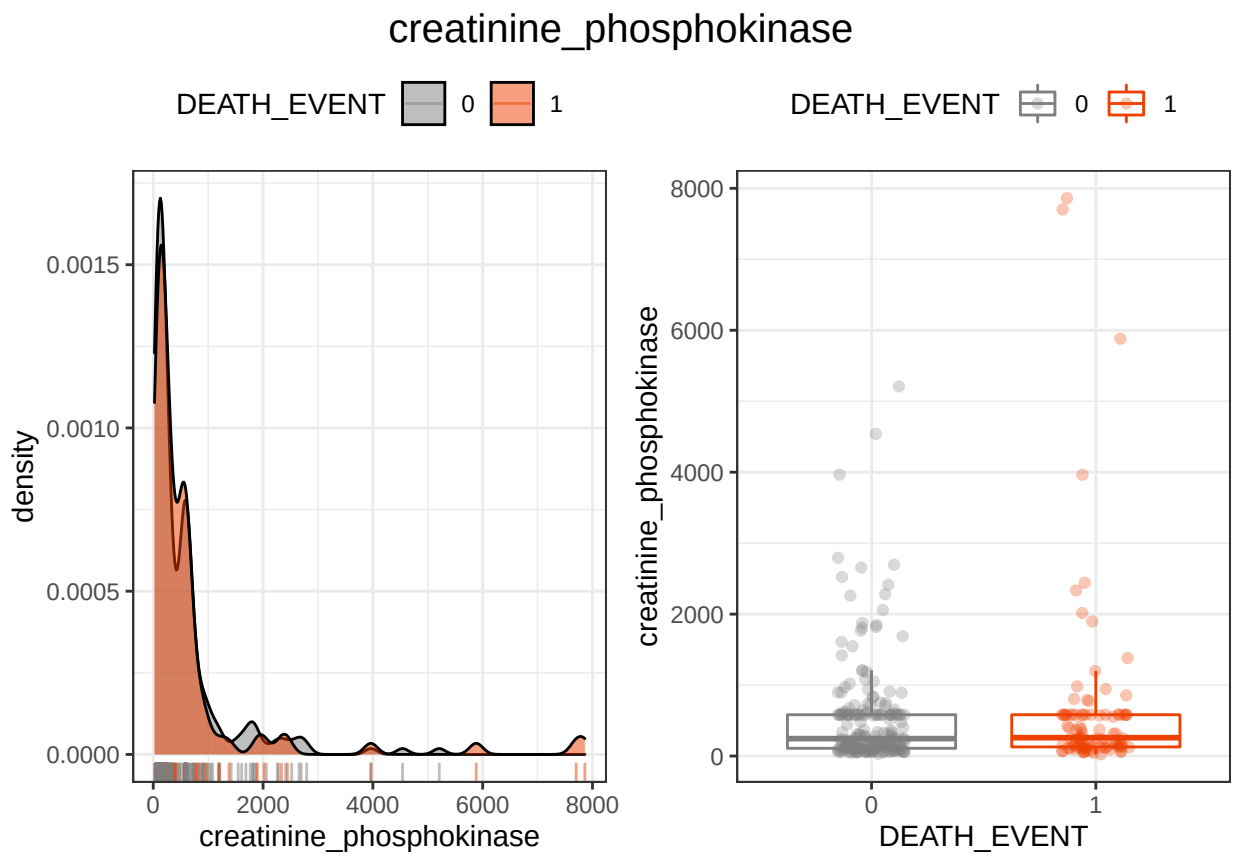
```
##   25%
## -1280
```

Si bien existen valores extremos para esta variable no existen criterios objetivos para eliminar dichas medidas por lo que mantendremos la totalidad de los valores.

Análisis de la variable frente a la variable dependiente:

```r
par(mfrow=c(1,2))
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:

```
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
```

No se observa que la variable sea discriminante.

ejection_fraction

Porcentaje de sangre que sale del corazón en cada contracción.

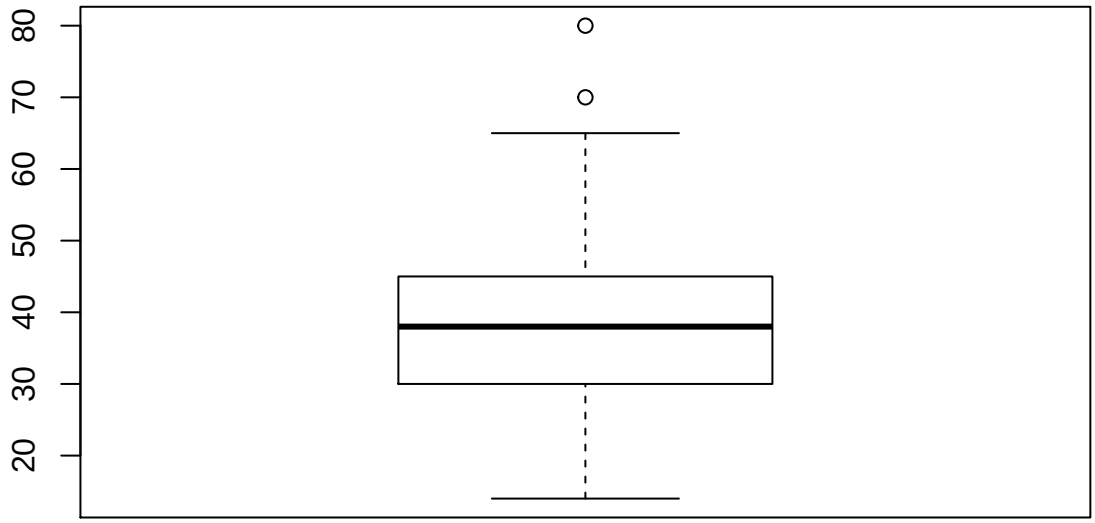
Estadísticos de la variable:

```
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)
```

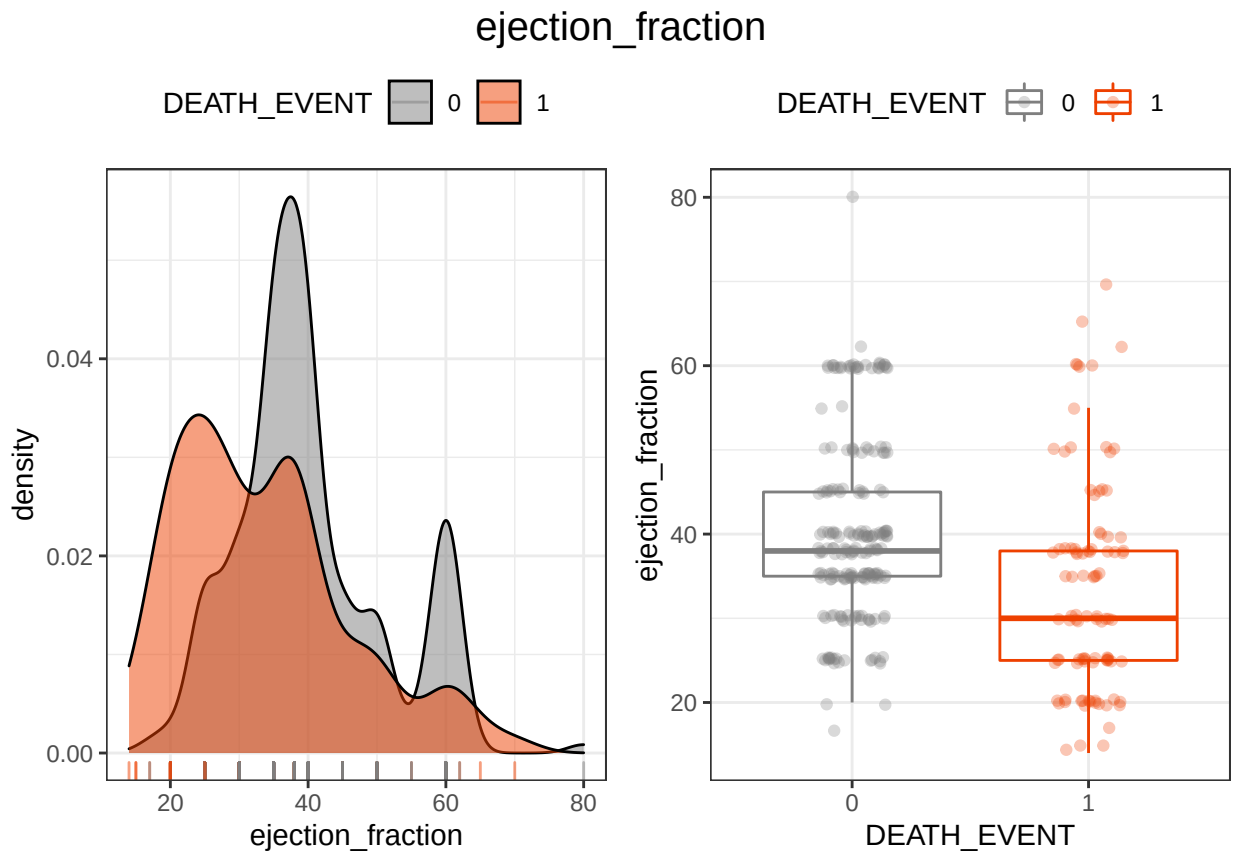


Se observan dos valores atípicos (outliers) pero no demasiado extremos.

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:

```
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
```

Se observa como para porcentajes por debajo del 30% la supervivencia es mayor.

platelets

Plaquetas en la sangre (kiloplatelets/mL).

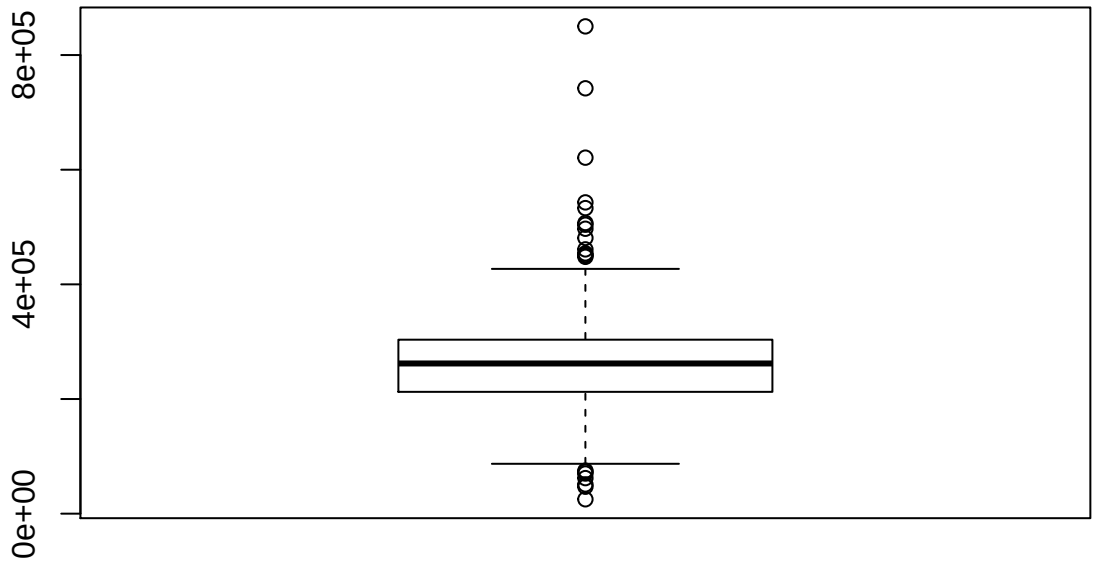
Estadísticos de la variable:

```
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).

Vamos a analizar cuales son los valores atípicos de la muestra utilizando el test de Tukey como hemos realizado anteriormente.

```
#Calculo de los cuartiles
qrts <- quantile(hf$platelets, probs = c(0.25, 0.75))

#Rango intercuartilico
iqr <- qrts[2]-qrts[1]
```

```r
# Umbral
h_leve <- 1.5 * iqr
h_extremo <- 3 * iqr

# Limite superior
limite_superior_leve <- qrts[2]+h_leve
limite_superior_extremo <- qrts[2]+h_extremo

# Limite inferior
limite_inferior_leve <- qrts[1]-h_leve
limite_inferior_extremo <- qrts[1]-h_extremo
```

- Limite superior leve:

```r
limite_superior_leve
```

```
##    75%
## 440000
```

- Limite superior extremo:

```r
limite_superior_extremo
```

```
##    75%
## 576500
```

- Limite inferior leve:

```r
limite_inferior_leve
```

```
##   25%
## 76000
```

- Limite inferior extremo:

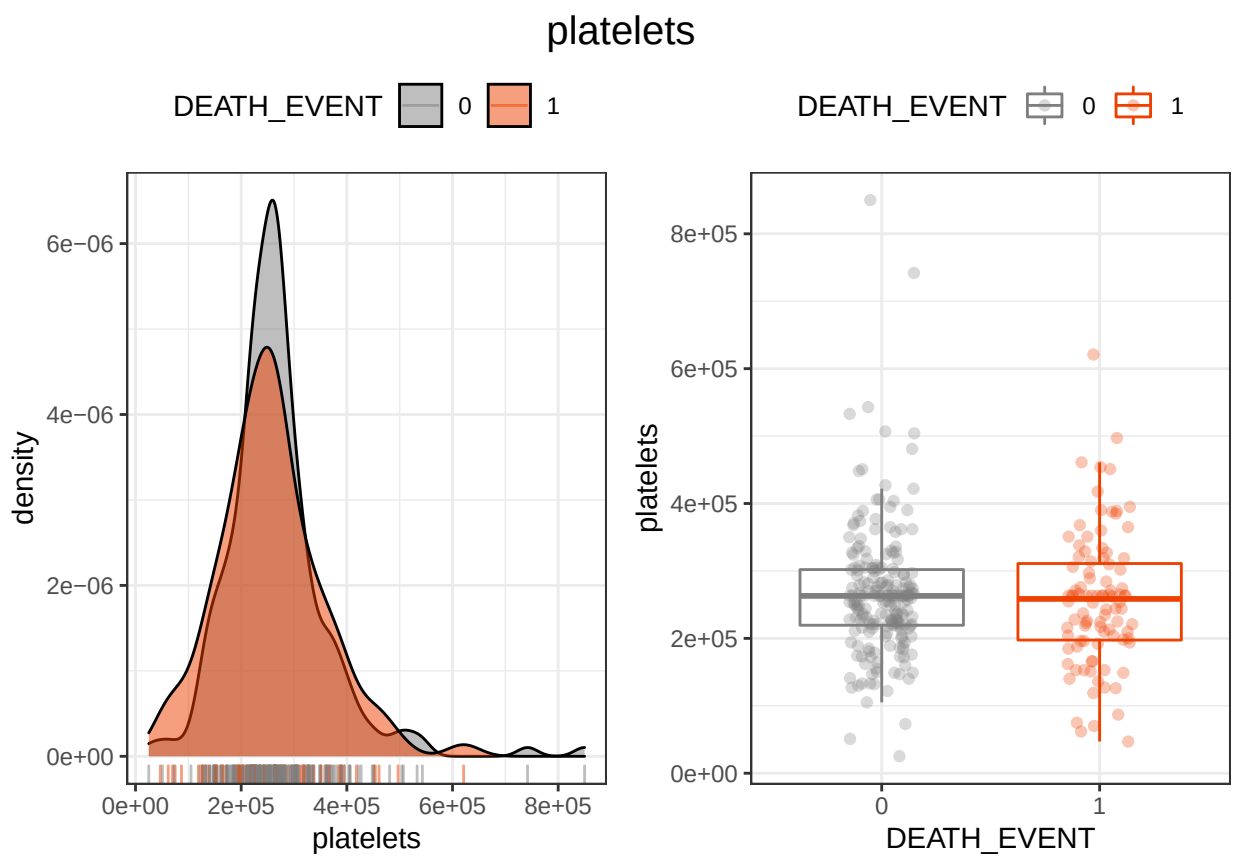
```r
limite_inferior_extremo
```

```
##    25%
## -60500
```

Si bien existen valores extremos para esta variable no existen criterios objetivos para eliminar dichas medidas por lo que mantendremos la totalidad de los valores.

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:

```
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
```

No se observa que la variable sea discriminante.

serum_creatinine

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

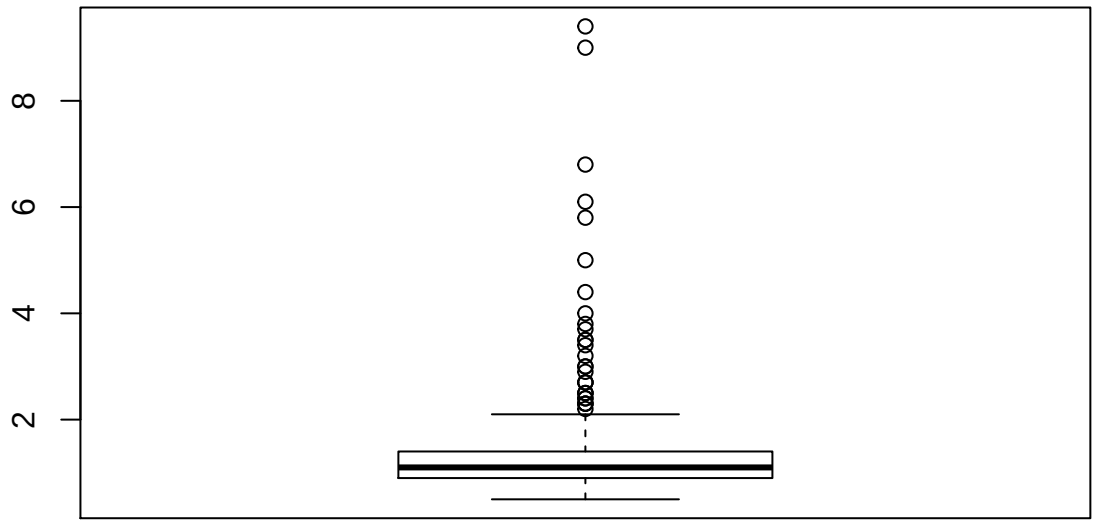
Estadísticos de la variable:

```
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).

Vamos a analizar cuales son los valores atípicos de la muestra utilizando el test de Tukey como hemos realizado anteriormente.

```r
#Calculo de los cuartiles
qrts <- quantile(hf$serum_creatinine, probs = c(0.25, 0.75))

#Rango intercuartilico
iqr <- qrts[2]-qrts[1]

# Umbral
h_leve <- 1.5 * iqr
h_extremo <- 3 * iqr

# Limite superior
limite_superior_leve <- qrts[2]+h_leve
limite_superior_extremo <- qrts[2]+h_extremo

# Limite inferior
limite_inferior_leve <- qrts[1]-h_leve
limite_inferior_extremo <- qrts[1]-h_extremo
```

- Limite superior leve:

```r
limite_superior_leve
```

```
##  75%
## 2.15
```

- Limite superior extremo:

```r
limite_superior_extremo
```

```
## 75%
## 2.9
```

- Limite inferior leve:

```r
limite_inferior_leve
```

```
##  25%
## 0.15
```

- Limite inferior extremo:

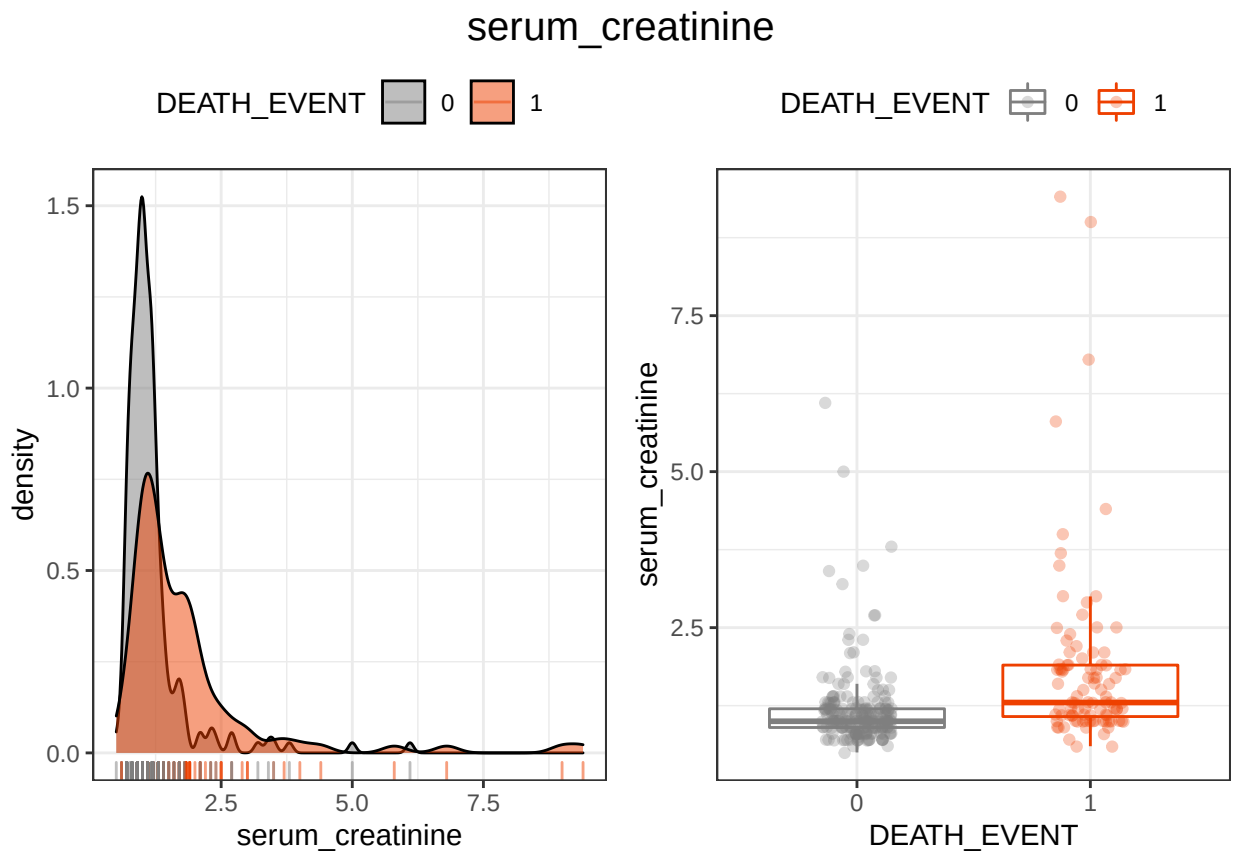
```r
limite_inferior_extremo
```

```
##  25%
## -0.6
```

Si bien existen valores extremos para esta variable no existen criterios objetivos para eliminar dichas medidas por lo que mantendremos la totalidad de los valores.

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:

```
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
```

Se observa que para niveles por encima mas/menos 1,5 el nivel de fallecimientos aumenta respecto a los supervivientes.

serum_sodium

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

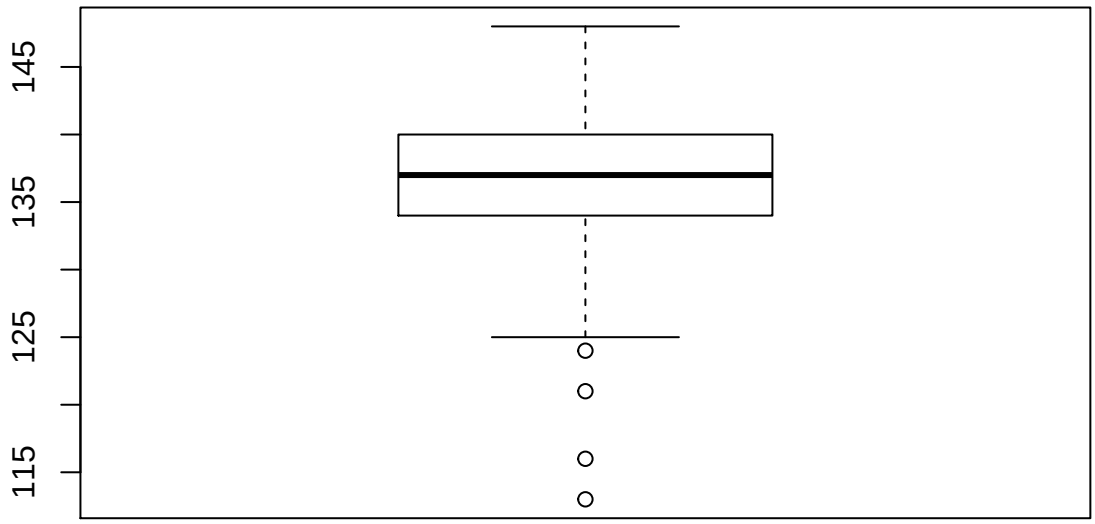
Estadísticos de la variable:

```
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).

Vamos a analizar cuales son los valores atípicos de la muestra utilizando el test de Tukey como hemos realizado anteriormente.

```r
#Calculo de los cuartiles
qrts <- quantile(hf$serum_sodium, probs = c(0.25, 0.75))

#Rango intercuartilico
iqr <- qrts[2]-qrts[1]

# Umbral
h_leve <- 1.5 * iqr
h_extremo <- 3 * iqr

# Limite superior
limite_superior_leve <- qrts[2]+h_leve
limite_superior_extremo <- qrts[2]+h_extremo

# Limite inferior
limite_inferior_leve <- qrts[1]-h_leve
limite_inferior_extremo <- qrts[1]-h_extremo
```

- Limite superior leve:

```r
limite_superior_leve
```

```
## 75%
## 149
```

- Limite superior extremo:

```r
limite_superior_extremo
```

```
## 75%
## 158
```

- Limite inferior leve:

```r
limite_inferior_leve
```

```
## 25%
## 125
```

- Limite inferior extremo:

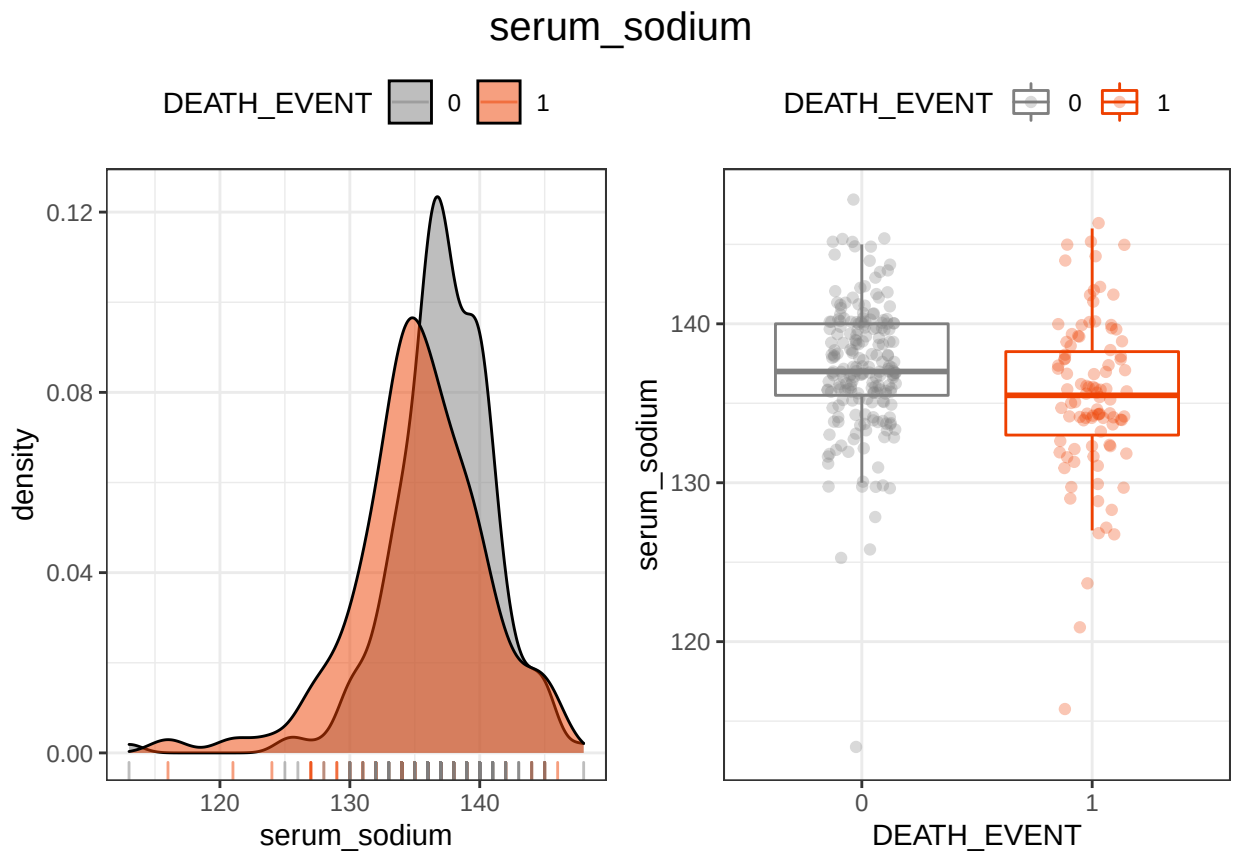
```r
limite_inferior_extremo
```

```
## 25%
## 116
```

Si bien existen valores extremos para esta variable no existen criterios objetivos para eliminar dichas medidas por lo que mantendremos la totalidad de los valores.

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:

```
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
```

Se observa que la variable que es dicriminante y que a mayor nivel hay mayor supervivencia.

time

Período de seguimiento (días).

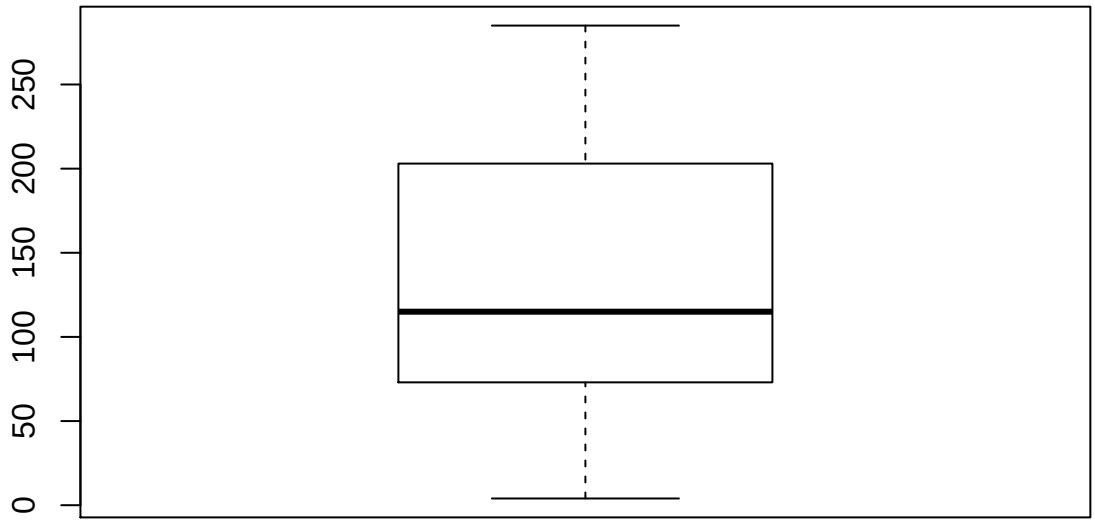
Estadísticos de la variable:

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

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

Boxplot:

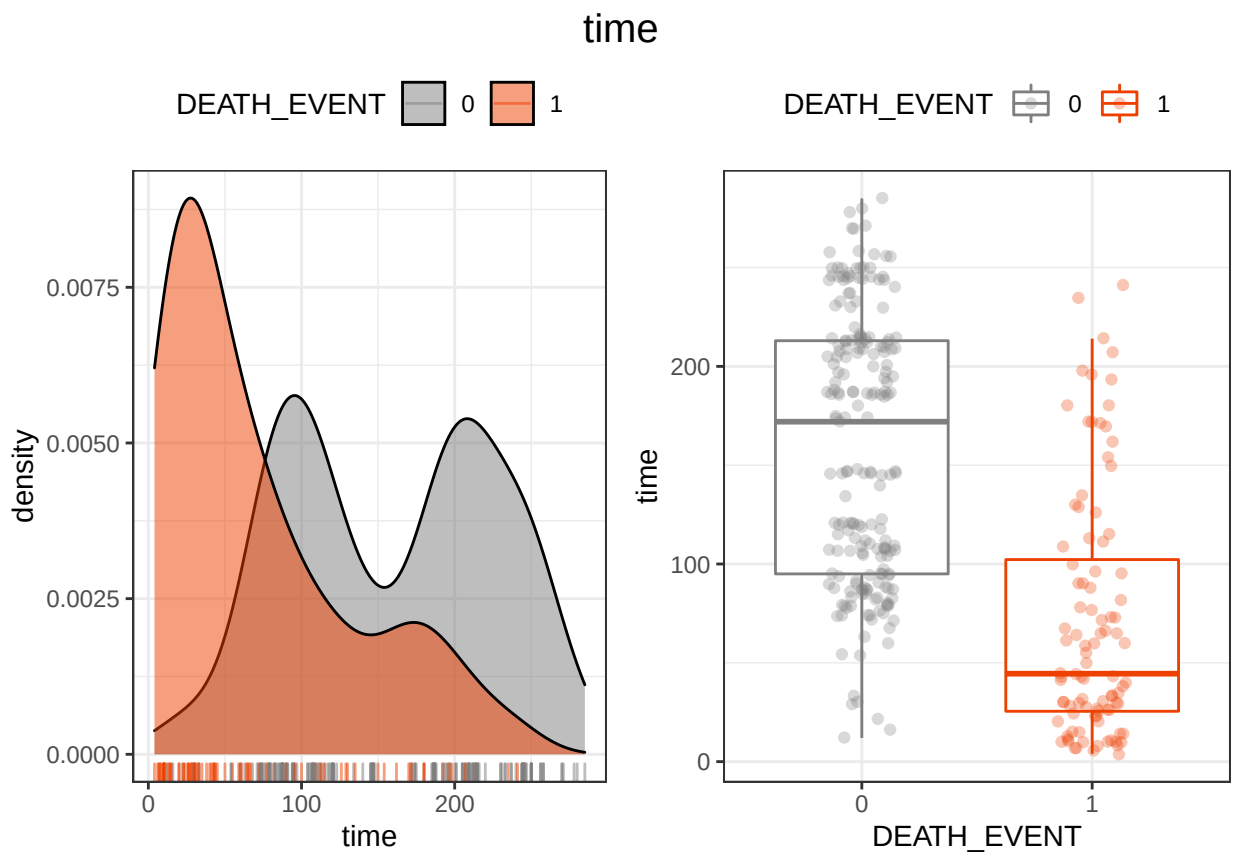
```
boxplot(hf$time)
```



No 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
```

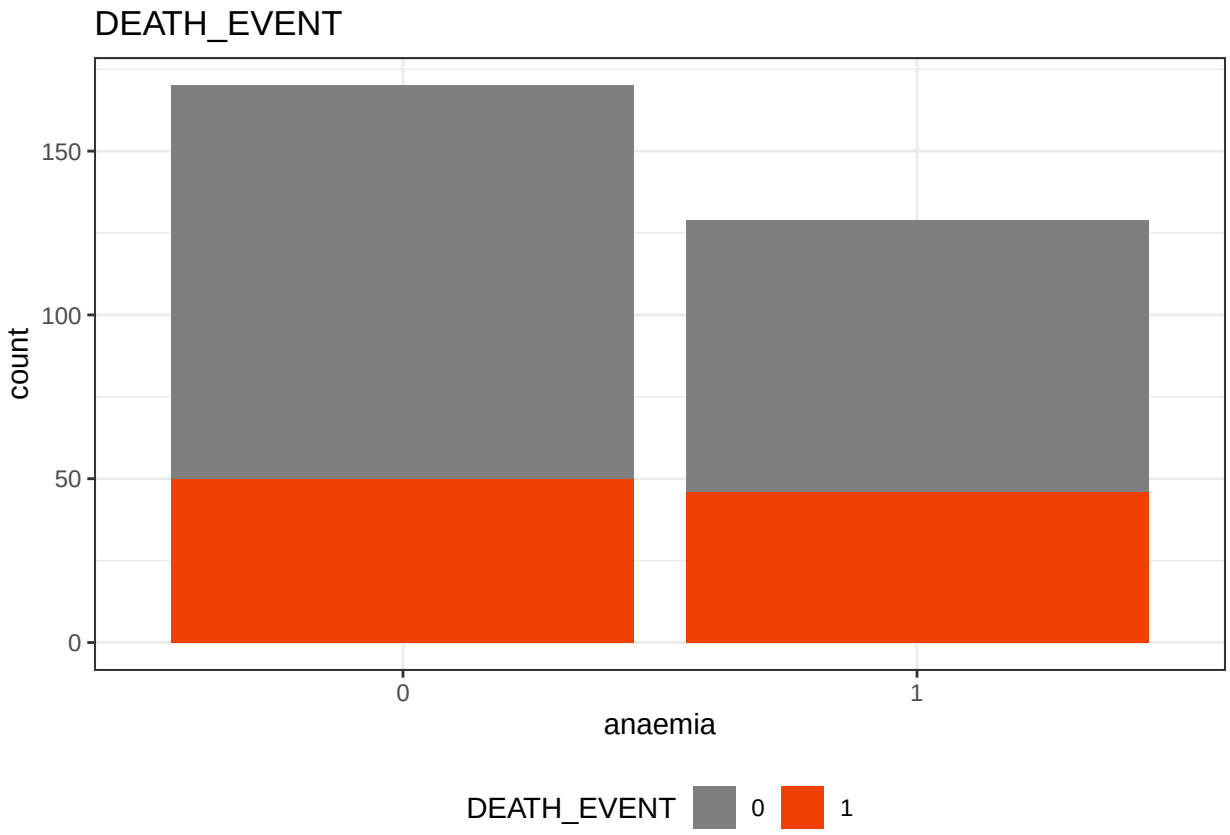
Esta variable refleja el tiempo que transcurre desde que el paciente es ingresado hasta que, o bién fallece o bien es dado de alta. No es una variable que pueda ser utilizada desde un punto de vista de diagnóstico pero sí puede ser utilizado como objetivo del análisis para estudiar si los valores medios en días de supervivencia de los pacientes vienen condicionados por algún tipo de variable.

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

anaemia

Si el paciente padece anemia (disminución de glóbulos rojos o hemoglobina).

```r
# factorizamos la variable

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

ggplot(data = hf, aes(x = anaemia, 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$anaemia, hf$DEATH_EVENT)
```

```
##
##       0   1
##   0 120  50
##   1  83  46
```

- Tabla de frecuencias (%):

```
prop.table(table(hf$anaemia, hf$DEATH_EVENT),1) %>% round(digits = 2)
```

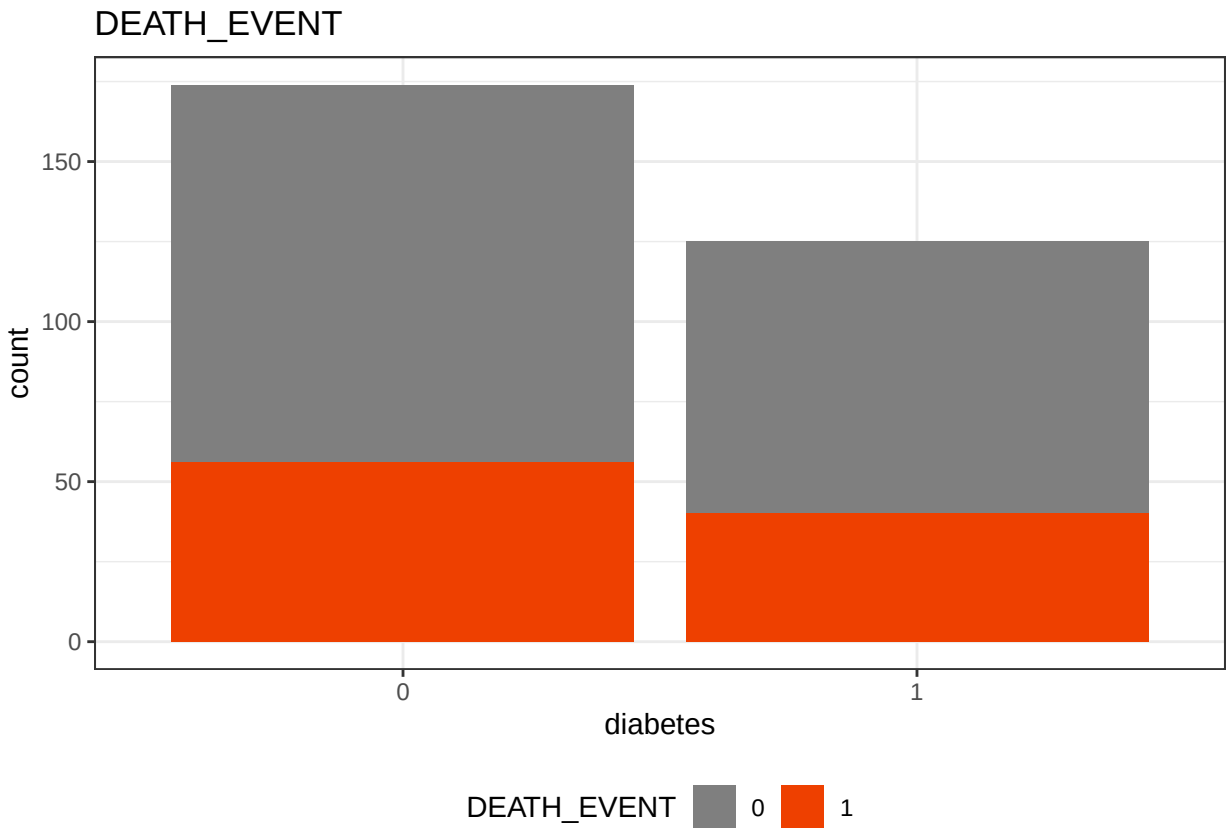
```
##
##        0    1
##   0 0.71 0.29
##   1 0.64 0.36
```

De análisis de los datos podría deducirse que el porcentaje de pacientes con anemia es mayor en aquellos en los que han fallecido. Aunque esta percepción deberá confirmarse estadísticamente mediante un contraste.

diabetes

Si el paciente padece diabetes.

```r
# factorizamos la variable

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

ggplot(data = hf, aes(x = diabetes, 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 (#):

```r
table(hf$diabetes, hf$DEATH_EVENT)
```

```
##
##        0   1
##    0 118  56
##    1  85  40
```

- Tabla de frecuencias (%):

```r
prop.table(table(hf$diabetes, hf$DEATH_EVENT),1) %>% round(digits = 2)
```

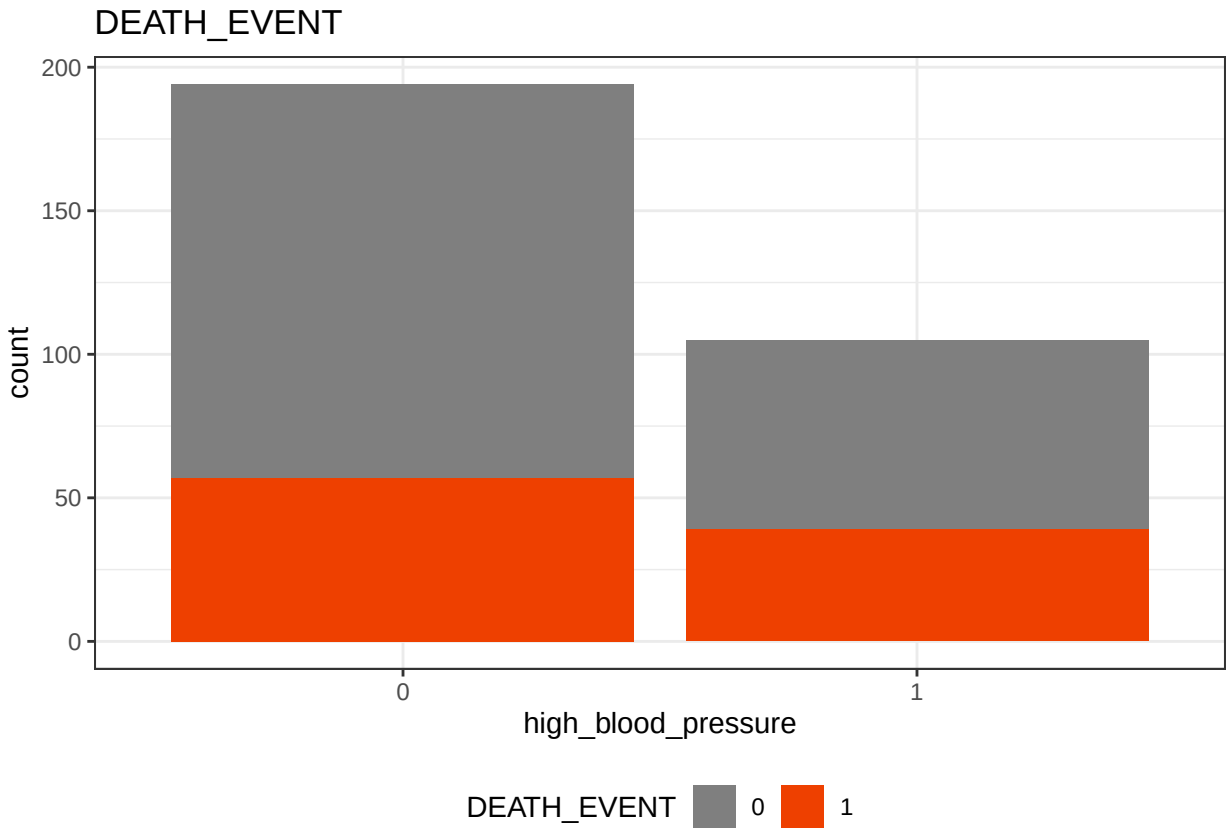
```
##
##        0    1
##   0 0.68 0.32
##   1 0.68 0.32
```

No se observa que la variable sea discriminante. Dado el número de fallecidos con diabetes es similar a los que no la tienen.

high_blood_pressure

Si el paciente padece hipertensión.

```r
# factorizamos la variable

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

ggplot(data = hf, aes(x = high_blood_pressure, 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 (#):

```r
table(hf$high_blood_pressure, hf$DEATH_EVENT)
```

```
##
##       0   1
##   0 137  57
##   1  66  39
```

- Tabla de frecuencias (%):

```r
prop.table(table(hf$high_blood_pressure, hf$DEATH_EVENT),1) %>% round(digits = 2)
```

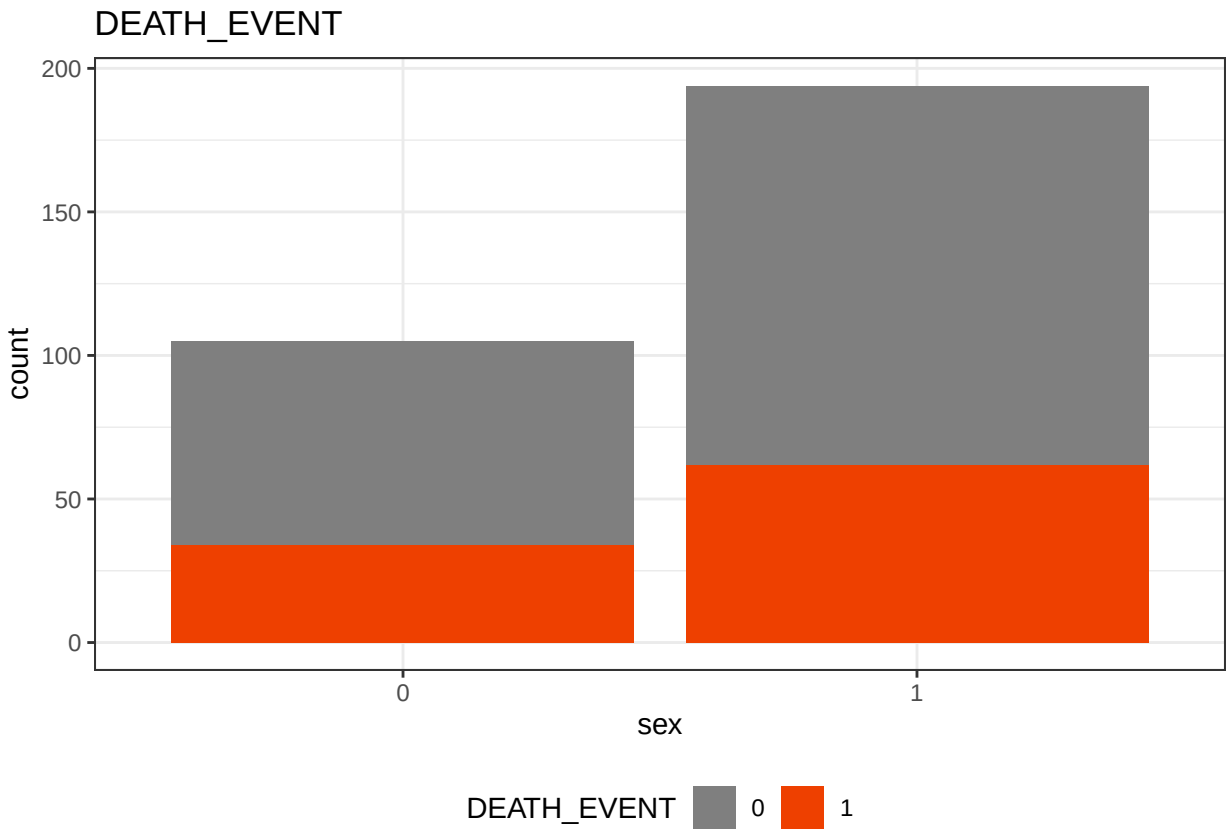
```
##
##        0    1
##   0 0.71 0.29
##   1 0.63 0.37
```

De análisis de los datos podría deducirse que el porcentaje de pacientes con hipertensión es mayor en aquellos en los que han fallecido. Aunque esta percepción deberá confirmarse estadísticamente mediante un contraste.

sex

Si el paciente es hombre.

```r
# factorizamos la variable

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

ggplot(data = hf, aes(x = sex, 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$sex, hf$DEATH_EVENT)
```

```
##
##        0   1
##   0  71  34
##   1 132  62
```

- Tabla de frecuencias (%):

```
prop.table(table(hf$sex, hf$DEATH_EVENT),1) %>% round(digits = 2)
```

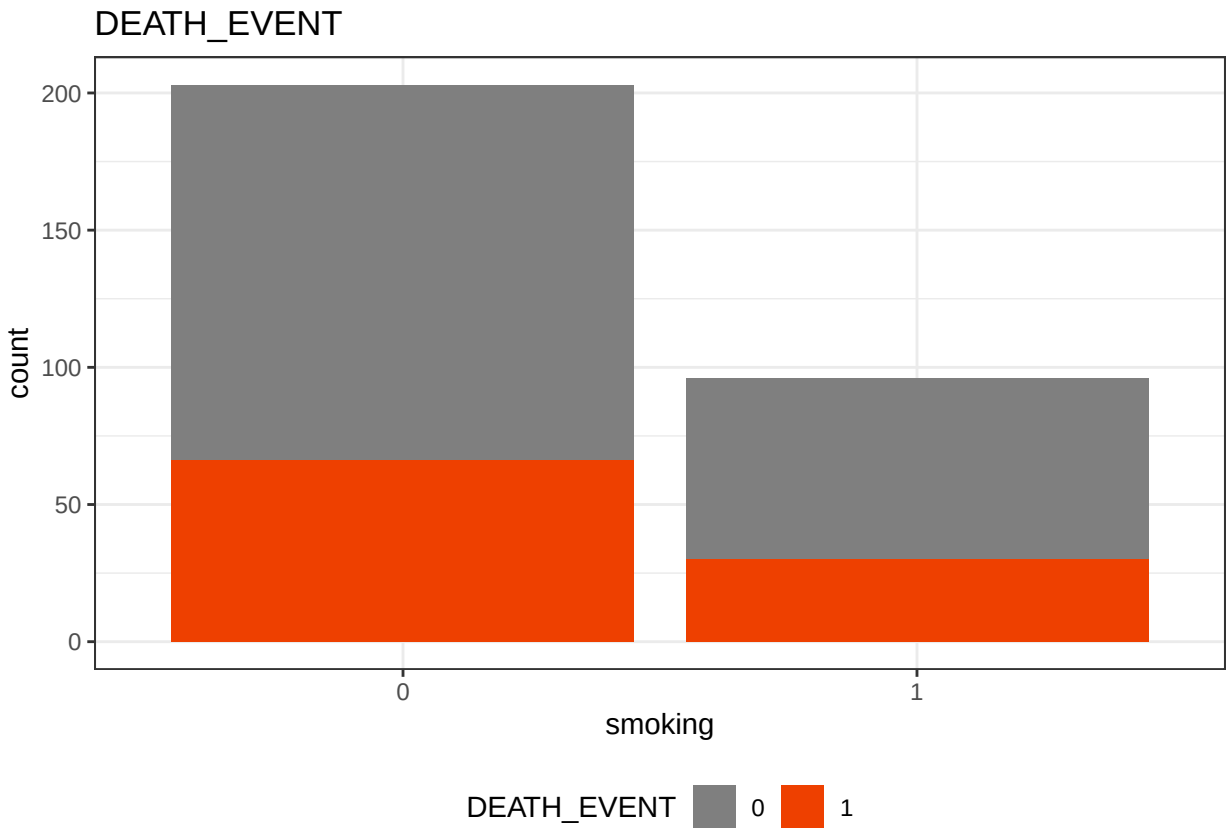
```
##
##        0    1
##   0 0.68 0.32
##   1 0.68 0.32
```

No se observa que la variable sea discriminante. Dado el número de fallecidos hombre y mujeres es similar a los que no la tienen.

smoking

Si el paciente es fumador.

```r
# factorizamos la variable

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

ggplot(data = hf, aes(x = smoking, 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 (#):

```r
table(hf$smoking, hf$DEATH_EVENT)
```

```
##
##       0   1
##   0 137  66
##   1  66  30
```

- Tabla de frecuencias (%):

```r
prop.table(table(hf$smoking, hf$DEATH_EVENT),1) %>% round(digits = 2)
```

```
##
##        0    1
##   0 0.67 0.33
##   1 0.69 0.31
```

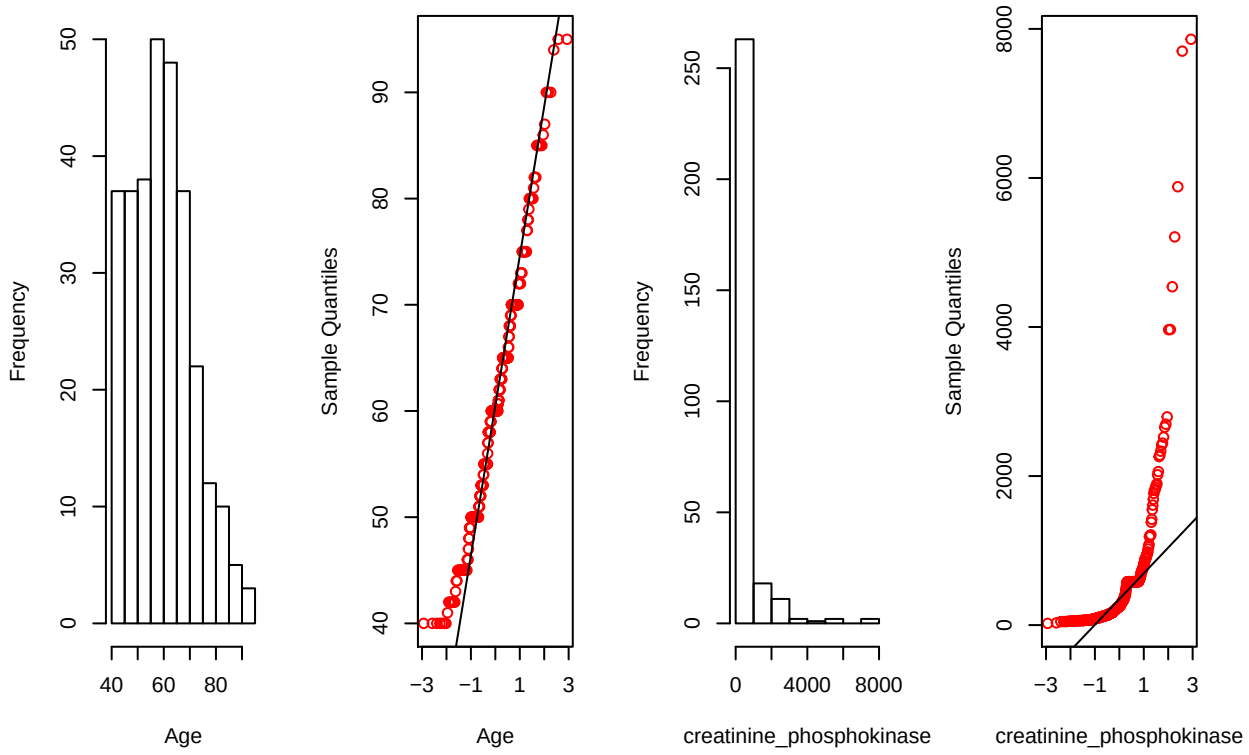
No se observa que la variable sea discriminante. Dado el número de fallecidos que fuman es similar a los que no fuman.

4.2. Comprobación de la normalidad de las variables

Comprobamos la normalidad de las variables mediante técnicas gráficas:

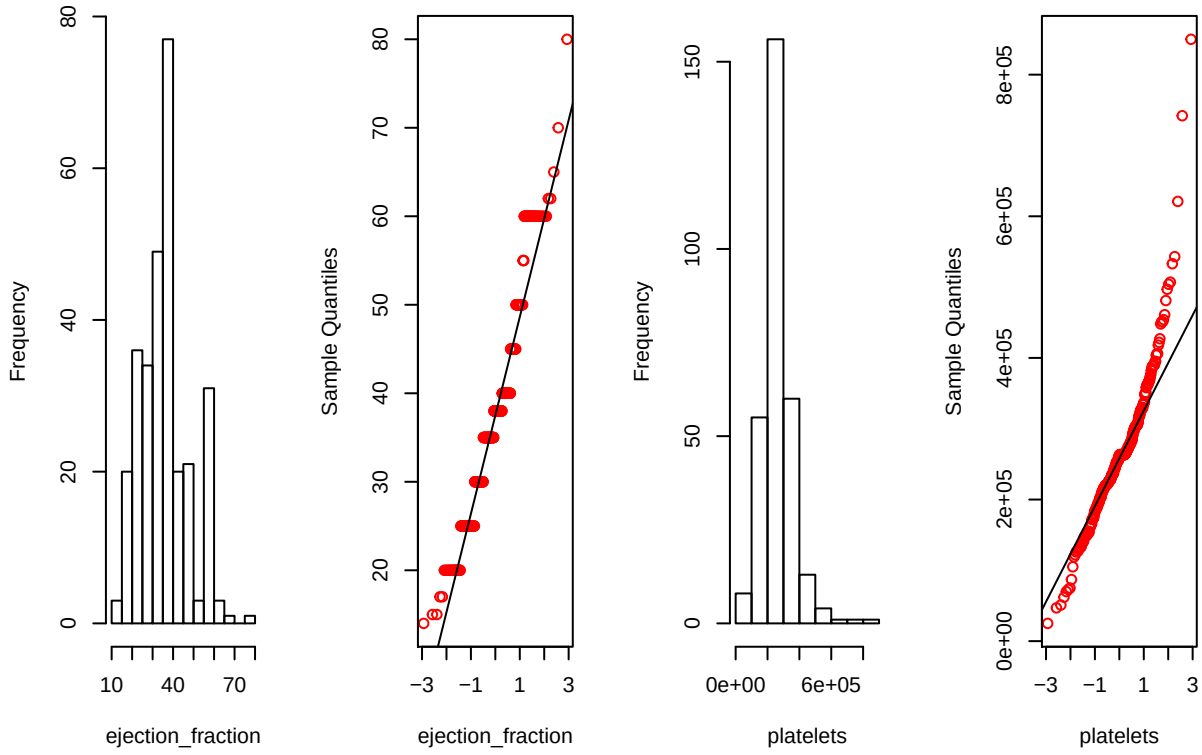
Age y creatinine_phosphokinase

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


hist(hf$age, main = "", xlab = "Age")
qqnorm(hf$age, main = "", xlab = "Age", col="red")
qqline(hf$age, main = "", xlab = "Age", col="black")

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



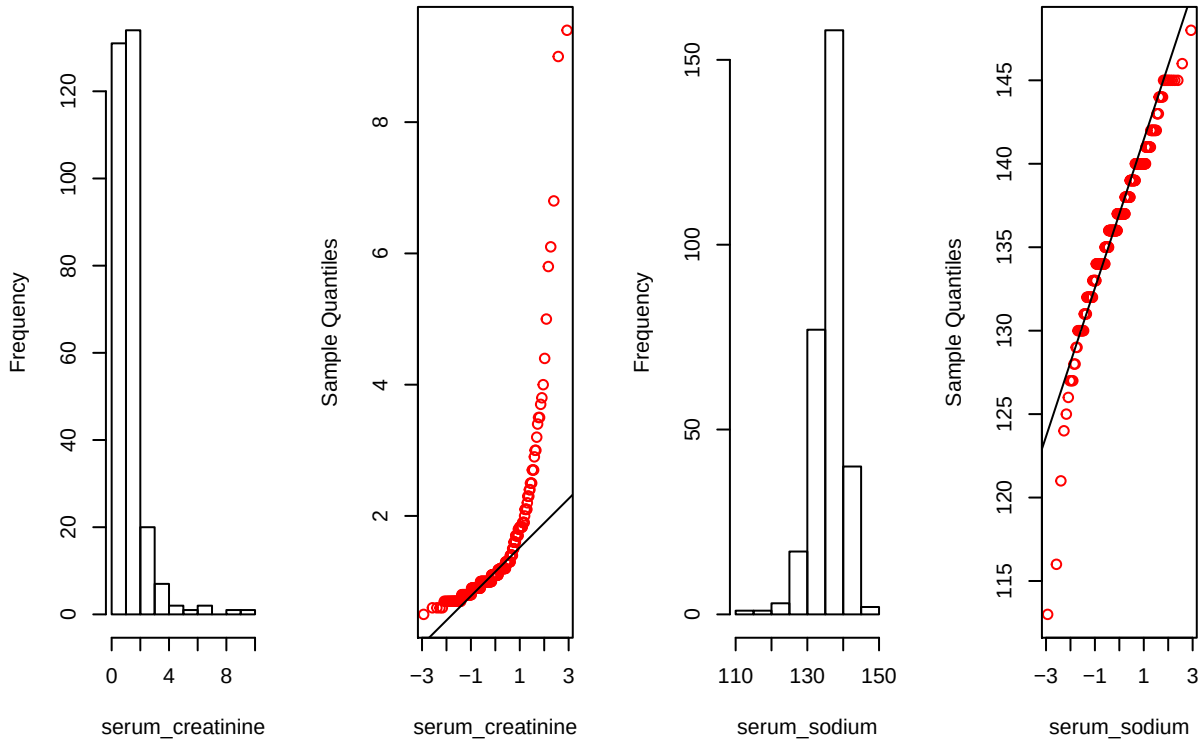
ejection_fraction y platelets

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

hist(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")

hist(hf$platelets, main = "", xlab = "platelets")
qqnorm(hf$platelets, main = "", xlab = "platelets", col="red")
qqline(hf$platelets, main = "", xlab = "platelets", col="black")
```



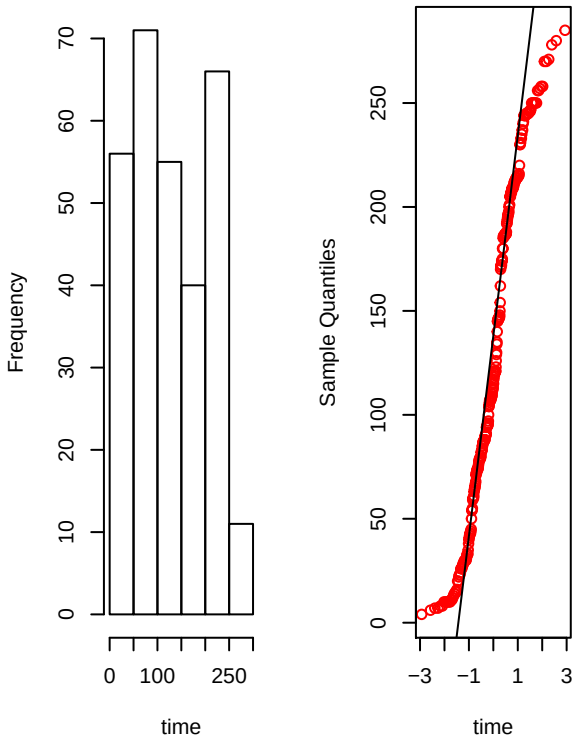
serum_creatinine y serum_sodium

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

hist((hf$serum_creatinine), main = "", xlab = "serum_creatinine")
qqnorm(hf$serum_creatinine, main = "", xlab = "serum_creatinine", col="red")
qqline(hf$serum_creatinine, main = "", xlab = "Age", col="black")

hist(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")
```



time

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

hist(hf$time, main = "", xlab = "time")
qqnorm(hf$time, main = "", xlab = "time", col="red")
qqline(hf$time, main = "", xlab = "Age", col="black")
```

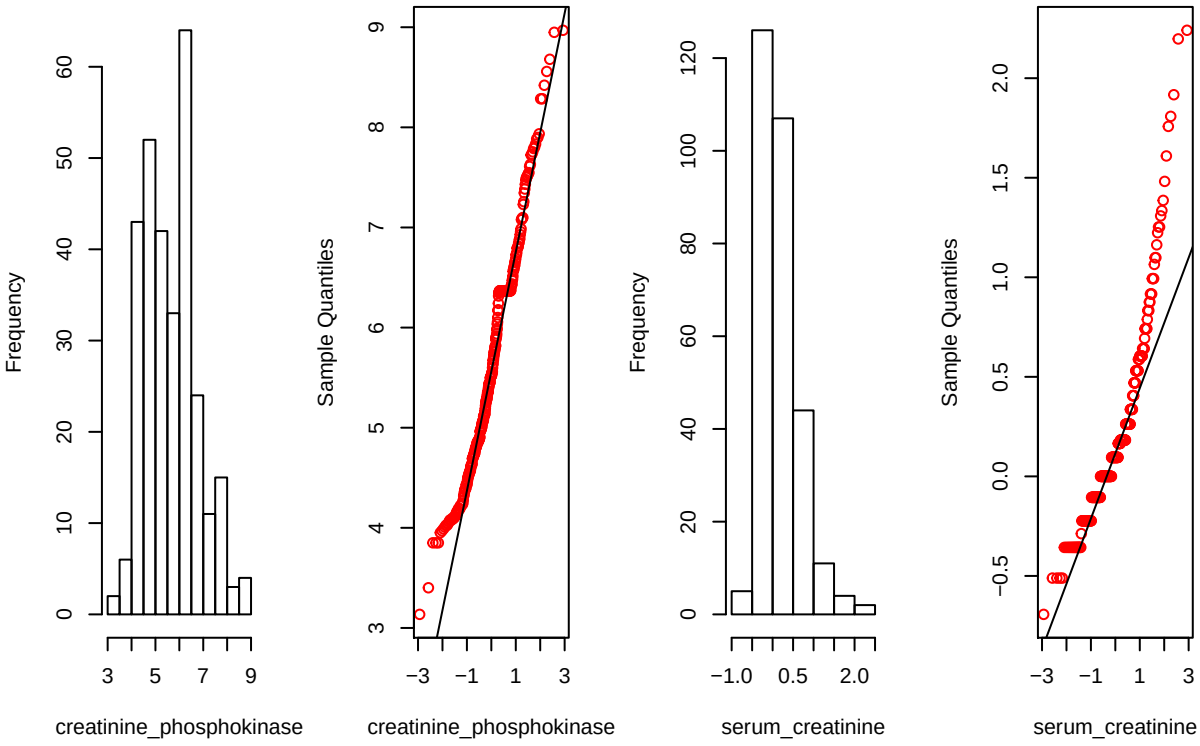


Age, ejection_fraction, platelets, serum_sodium y time son variables más o menos normales (puede observarse en la gráfica de distribución y en la linea qqnorm.

Creatinine_phosphokinase, serum creatinine no están normalizadas (con claramente logaritmicas)

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(1,4))

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


Frequency
creatinine_phosphokinase
Sample Quantiles
creatinine_phosphokinase
Frequency
serum_creatinine
Sample Quantiles
serum_creatinine

4.3. Análisis de correlación entre las variables numéricas:

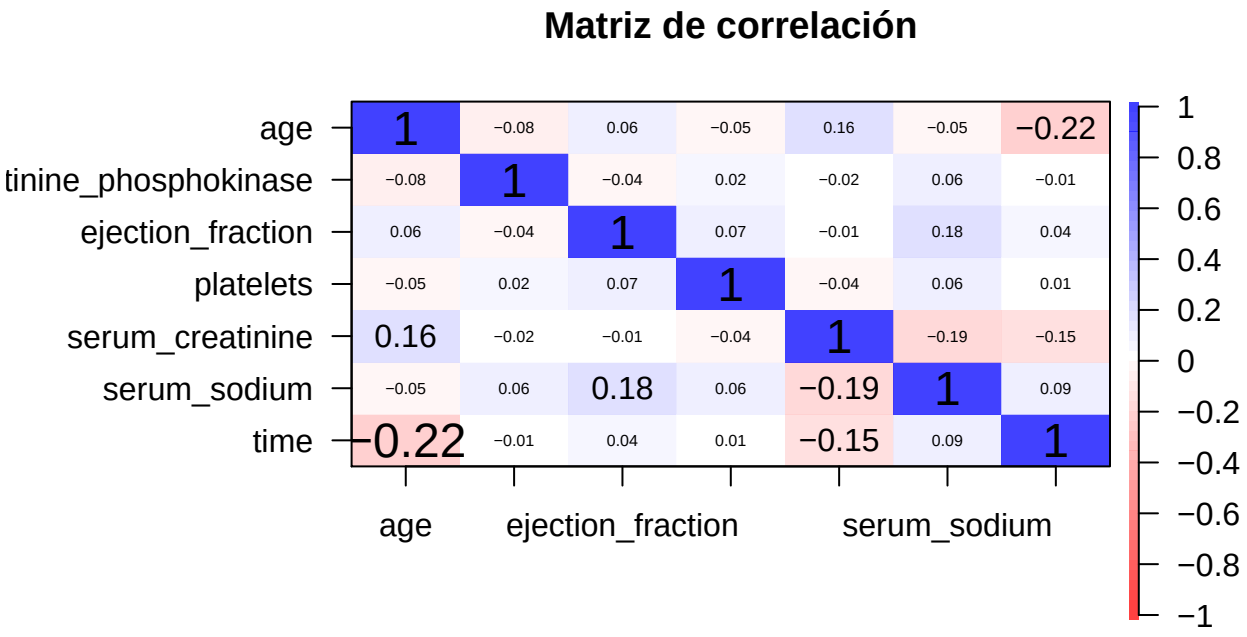
Vamos a analizar si existe correlación entre las variables numéricas.

```
# Seleccionamos las variables numéricas y las cargamos en un data set

cor <- select(hf,
              age,
              creatinine_phosphokinase,
              ejection_fraction,
              platelets,
              serum_creatinine,
              serum_sodium,
              time)


cor(cor)
```

```
##                              age creatinine_phosphokinase ejection_fraction
## age                   1.00000000             -0.081583900        0.06009836
## creatinine_phosphokinase -0.08158390              1.000000000       -0.04407955
## ejection_fraction      0.06009836             -0.044079554        1.00000000
## platelets             -0.05235437              0.024463389        0.07217747
## serum_creatinine       0.15918713             -0.016408480       -0.01130247
## serum_sodium          -0.04596584              0.059550156        0.17590228
## time                  -0.22406842             -0.009345653        0.04172924
##                            platelets serum_creatinine serum_sodium         time
## age                   -0.05235437       0.15918713  -0.04596584 -0.224068420
## creatinine_phosphokinase  0.02446339      -0.01640848   0.05955016 -0.009345653
## ejection_fraction      0.07217747      -0.01130247   0.17590228  0.041729235
## platelets              1.00000000      -0.04119808   0.06212462  0.010513909
## serum_creatinine      -0.04119808       1.00000000  -0.18909521 -0.149315418
## serum_sodium           0.06212462      -0.18909521   1.00000000  0.087640000
## time                   0.01051391      -0.14931542   0.08764000  1.000000000
```

- Gráfico con la matriz de correlación:

```
corPlot(cor, cex = 1, main = "Matriz de correlación")
```



No se aprecia que exista correlación entre las diferentes variables numéricas.

5.- ANALISIS

Dado el análisis de la información realizado sobre las variables nos planteamos tres análisis a realizar sobre los datos del dataset:

1.- Análisis de contraste respecto a si la media en días que tarda un paciente en morir depende de las condiciones que se presentan en variables no numéricas: si es fumador o no, el sexo, si tiene o no amemia, diabetes o presión elevada.

2.- Análisis de contraste respecto a si la probabilidad de morir depende de las condiciones que se presentan en variables no numéricas: si es fumador o no, el sexo, si tiene o no amemia, diabetes o presión elevada.

3.- Regresión logística para calcular si hay más probabilidad de morir que no en función de los parámetros médicos a la recepción del paciente.

5.1.- CONTRASTE DE HIPOTESIS.

Segmentamos el datastet

```
#Creación de subsets de trabajo de variables cualitativas anaemia, diabetes,
#high_blood_pressure


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

5.1.1 - CONTRASTE SOBRE LA MEDIA EN EL TIEMPO EN FALLECER

Para la realización del contraste necesitamos calcular previamente la Media del tiempo en fallecer para cada subset creado.

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

# fumadores/no fumadores
sm <-hf_sm$time[hf_sm$DEATH_EVENT==1]
nosm <-hf_nosm$time[hf_nosm$DEATH_EVENT==1]

# hombre/mujer
m <-hf_m$time[hf_m$DEATH_EVENT==1]
w <-hf_w$time[hf_w$DEATH_EVENT==1]

# anemico/no anemico
a <-hf_a$time[hf_a$DEATH_EVENT==1]
noa <-hf_noa$time[hf_noa$DEATH_EVENT==1]

#diabetico/ no diabetico
```

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

# hipertenso/no hipertenso
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
```

A priori podríamos suponer que el tiempo medio que tarda en fallecer una persona sí depende de alguna de las variables cualitativas, pero realizamos el contraste estadístico de cara aconfirmar tales suposiciones.

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


# fumadores/no fumadores
vsm<-var.test(sm,nosm, conf.level=0.95)
# sexo
vm<-var.test(m, w, conf.level=0.95)
#anémicos/no anemicos
va<-var.test(a,noa, conf.level=0.95)
# diabeticos/no diabeticos
vd<-var.test(d,nod, conf.level=0.95)
# hipertensos/no hipertensos
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, dado que son muestras de 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

Por defecto trabajaremos con un nivel de confianza del 95% y por lo tanto un nivel de significación del 5% (0.05)

```r
#Contraste de hipótesis. Muestras poblacionales. Varianzas desconocidas pero iguales.
#Con un 95% de nivel de confianza

#fumadores/no fumadores
test_sm<-t.test(sm,nosm, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
#sexo
test_m<-t.test(m, w, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
#anémico/no anémico
test_a<-t.test(a,noa, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
#diabetico/no diabetico
test_d<-t.test(d,nod, alternative="two.sided", var.equal=TRUE, conf.level=0.95)
#hipertenso/no hipertenso
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). No pueden rechazarse ninguna de las hipótesis nulas y por lo tanto las medias de tiempo no son diferentes en ningún caso.

5.1.2 - CONTRASTE SOBRE LOS PORCENTAJES DE FALLECIDOS Calculamos los porcentajes de fallecidos frente a los totales de cada muestra

```
# Estudio de porcentajes respecto a DEATH_EVENT

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

# fumadores/no fumadores
psm<- nrow(hf_sm[hf_sm$DEATH_EVENT==1,])/nrow(hf_sm)
pnosm<- nrow(hf_nosm[hf_nosm$DEATH_EVENT==1,])/nrow(hf_nosm)

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

# anemicos/no amemicos
pa <- nrow(hf_a[hf_a$DEATH_EVENT==1,])/nrow(hf_a)
pnoa <- nrow(hf_noa[hf_noa$DEATH_EVENT==1,])/nrow(hf_noa)

# diabeticos/no diabeticos
pd <- nrow(hf_d[hf_d$DEATH_EVENT==1,])/nrow(hf_d)
pnod <- nrow(hf_nod[hf_nod$DEATH_EVENT==1,])/nrow(hf_nod)

# hipertensos/no hipertensos
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 en fumadores/no fumadores
c(psm,pnosm)
```

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

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

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

```
#porcentajes por anemicos/no anemicos
c(pa,pnoa)
```

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

```
#porcentajes por diabeticos/no diabeticos
c(pd,pnod)
```

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

```
#porcentajes por hipertensos/no hipertensos
c(php,pnohp)
```

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

Aunque los porcentajes ya se ven muy similares, 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

```r
# Contraste de hipótesis para porcentajes


# fumador/no fumador
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")

#anemico/no anémico
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")

#diabetico/no diabetico
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")

#hipertenso/no hipertenso
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.

5.2 - MODELO DE REGRESION LOGISTICA. DIAGNOSTICO DE PROBABILIDAD DE FALLECIMIENTO TRAS UN INFARTO

Para obtener un modelo que sirva para diagnosticar las probabilidades de supervivencia tras un infarto vamos a calcular un modelo de regresión logístico. Esto es así dado que la variable a predecir es dicotómica. Podríamos utilizar otros modelos de predicción (selección, clasificación) pero vamos a realizar un modelo logístico.

Calcularemos inicialmente las variables que más intervienen en el modelo y posteriormente obtendremos una estimación de la seguridad del mismo.

Las variables serum_creatinine y creatinine_phosphokinase las vamos a utilizar en forma logarítmica.

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

```
# modelo logístico

glm_hf1 <- glm(formula= factor(DEATH_EVENT)~ age + log(serum_creatinine) +
                 ejection_fraction + log(creatinine_phosphokinase) + serum_sodium +
                 platelets + factor(sex) + factor(smoking) +
                 factor(high_blood_pressure) + factor(diabetes) + anaemia ,
                 family=binomial(link=logit), data = hf)
summary(glm_hf1)
```

```
##
## Call:
## glm(formula = factor(DEATH_EVENT) ~ age + log(serum_creatinine) +
##     ejection_fraction + log(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
## -2.0752  -0.7504  -0.4412   0.7893   2.5954
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   2.502e+00  4.588e+00   0.545 0.585517
```

```
## age                            4.979e-02  1.307e-02   3.809 0.000139 ***
## log(serum_creatinine)          1.636e+00  3.460e-01   4.729 2.26e-06 ***
## ejection_fraction             -6.403e-02  1.446e-02  -4.430 9.44e-06 ***
## log(creatinine_phosphokinase)  2.411e-01  1.350e-01   1.785 0.074255 .
## serum_sodium                  -4.263e-02  3.317e-02  -1.285 0.198701
## platelets                     -7.139e-07  1.611e-06  -0.443 0.657647
## factor(sex)1                  -3.976e-01  3.520e-01  -1.130 0.258639
## factor(smoking)1               1.800e-01  3.526e-01   0.510 0.609750
## factor(high_blood_pressure)1   5.030e-01  3.098e-01   1.623 0.104493
## factor(diabetes)1              9.947e-02  2.992e-01   0.332 0.739559
## anaemia1                       4.721e-01  3.069e-01   1.539 0.123884
## ---
## 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: 291.75  on 287  degrees of freedom
## AIC: 315.75
##
## Number of Fisher Scoring iterations: 5
```

Interpretación modelo logístico:

El logit, es decir, ln(P(Y=1/X)/1-P(Y=1/X)) es 2.502 + 4.979e-02 * age + 1.636e+00 * log(serum_creatinine) -6.403e-02 * ejection_fraction + 2.411e-01 * 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).

Ninguna de las variables dicotómicas tienen peso en el modelo.

AIC (Akaike Information Criterion) es el metodo matemático utilizado para calcular cómo de bueno es un modelo. Cuanto menor sea el valor de AIC mejor será el modelo.

Se pueden calcular los Odds Ratio de cada variable como el exp de los coeficientes (por defecto con un nivel de confianza del 95%).

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

```
##                   (Intercept)                           age
##                    12.2047616                     1.0510525
##         log(serum_creatinine)             ejection_fraction
##                     5.1352337                     0.9379772
## log(creatinine_phosphokinase)                  serum_sodium
##                     1.2726028                     0.9582640
##                     platelets                   factor(sex)1
##                     0.9999993                     0.6719065
##              factor(smoking)1  factor(high_blood_pressure)1
##                     1.1972025                     1.6536619
##             factor(diabetes)1                      anaemia1
##                     1.1045874                     1.6034372
```

El odds ratio se relaciona con la probabilidad cruzada entre la variable independiente y la variable dependiente.

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 1.051. Es decir la probabilidad de fallecer dividido por la probabilidad de no fallecer es un 1.051 mayor.

De cara a conocer cuáles son las variables a tener en cuenta en el modelo logístico hay que tener en cuenta, por tanto, el valor del odd ratio, el p valor y el AIC del modelo resultante.

Creamos varios modelos y seleccionamos el que tiene el AIC menor

```
# Determinación del modelo logístico

# en primer lugar con las variables numéricas significativas desde el punto de vista del
# p-value del estadístico de Wald
glm_hf2 <- glm(formula= factor(DEATH_EVENT)~ age + log(serum_creatinine) +
                 ejection_fraction, family=binomial(link=logit), data = hf)
glm_hf2 [["aic"]]
```

```
## [1] 310.0296
```

```
#vamos añadiendo variables numéricas de menor a mayor significancia
glm_hf3 <- glm(formula= factor(DEATH_EVENT)~ age + log(serum_creatinine) +
               ejection_fraction + log(creatinine_phosphokinase),
               family=binomial(link=logit), data = hf)
glm_hf3 [["aic"]]
```

```
## [1] 310.6846
```

```
glm_hf4 <- glm(formula= factor(DEATH_EVENT)~ age + log(serum_creatinine) +
               ejection_fraction  + serum_sodium , family=binomial(link=logit),
               data = hf)
glm_hf4 [["aic"]]
```

```
## [1] 310.677
```

```
#añadimos ahora variables dicotómicas a partir del mejor modelo con variables numéricas
glm_hf5 <- glm(formula= factor(DEATH_EVENT)~ age + log(serum_creatinine) +
               ejection_fraction +  high_blood_pressure,
               family=binomial(link=logit), data = hf)
glm_hf5 [["aic"]]
```

```
## [1] 309.0698
```

```
glm_hf6 <- glm(formula= factor(DEATH_EVENT)~ age + log(serum_creatinine) +
                 ejection_fraction + serum_sodium + high_blood_pressure +
                 anaemia , family=binomial(link=logit), data = hf)
glm_hf6 [["aic"]]
```

```
## [1] 310.2929
```

El modelo que proporciona el AIC más bajo es el numero 5 que contempla las variables age, log(serum_creatinine), ejection_fraction y high_blood_pressure.

Obtenemos la precisión del modelo mediante la metriz de confusión. Utilizamos la que aporta el paquete caret de R.

```
# Precisión del modelo estimado. Matriz de confusión

confusionMatrix(table(predict(glm_hf5, type="response") >= 0.5,hf$DEATH_EVENT == 1))
```

```
## Confusion Matrix and Statistics
##
##
##         FALSE TRUE
##   FALSE   181   48
##   TRUE     22   48
##
##                Accuracy : 0.7659
##                  95% CI : (0.7137, 0.8127)
##     No Information Rate : 0.6789
##     P-Value [Acc > NIR] : 0.0006002
##
##                   Kappa : 0.4217
##
##  Mcnemar's Test P-Value : 0.0028074
##
##             Sensitivity : 0.8916
##             Specificity : 0.5000
##          Pos Pred Value : 0.7904
##          Neg Pred Value : 0.6857
##              Prevalence : 0.6789
##          Detection Rate : 0.6054
##    Detection Prevalence : 0.7659
##       Balanced Accuracy : 0.6958
##
##        'Positive' Class : FALSE
##
```

El modelo tiene una precisión del 76.6 %, una sensibilidad del 50% (capacidad de predecir fallecimientos) y una especificidad del 89% (capacidad de estimar supervivientes).

Nota: la función de R muestra una sensibilidad del 89% y una especificidad del 50% pero está considerando como 'Positivos' a la clase FALSE (al contrario que estamos buscando).

6.- CONCLUSIONES

Se ha realizado un análisis pormenorizado de cada variable del dataset, con mayor foco en la variable a discriminar. Este análisis visual ya nos informa de que las variables dicotómicas iban a tener poca capacidad predictiva o estadística.

Se ha realizado un analisis de l normalidad de las variables numéricas y se han normalizado aquellas que presentaban menor forma normal mediante la utilización del logaritmo de las mismas.

Se ha realizado una serie de contrastes estadísticos con dos preguntas a responder:

1) ¿Depende de factores como la anemia, hipertensión, sexo, ser fumador o no o diabetes para tener mayor tiempo medio de supervivencia en caso de un ataque al corazón?, La respuesta es NO. Ninguno de estos factores es significativo con un nivel de confianza del 95%.

2) ¿Depende de factores como la anemia, hipertensión, sexo, ser fumador o no o diabetes para tener mayor probabilidad de fallecer en caso de un ataque al corazón? La respuesta también es NO. Ninguno de estos factores es significativo con un nivel de confianza del 95%.

Posteriormente se ha realizado un modelo de regresión logística a partir de los datos (dado que la variable a predecir es dicotómica) para predecir si un paciente en el momento de producirse un ataque al corazón tiene más probabilidad o no (fijándola en el 50%) de que fallezca.

El modelo de regresión logística nos indica que las variables más importantes que intervienen en el modelo son age, log(serum_creatinine), ejection_fraction y high_blood_pressure . Esta última en muy pequeña proporción.

El modelo desarrollado tiene una fiabilidad de casi un 77% con un 50% de capacidad de discernimiento en pacientes con probabilidad de fallecer.

7.- BIBLIOGRAFIA

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.
Guillén Estany, Montserrat; Alonso Alonso, María Teresa (2020) Modelos de Regresión Logística. UOC.

8.- AGRADECIMIENTOS DATASET

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). https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-1023-5

License

9.- ANEXO

9.1. Estudio de la importacia de las variables en los modelos.

Se podría hacer un anális rápido para seleccionar las variables con las que trabajar. Para ello hemos seleccionado el paquete Boruta dada su capacidad de mostrar visualmente la importancia de cada variable del dataset.

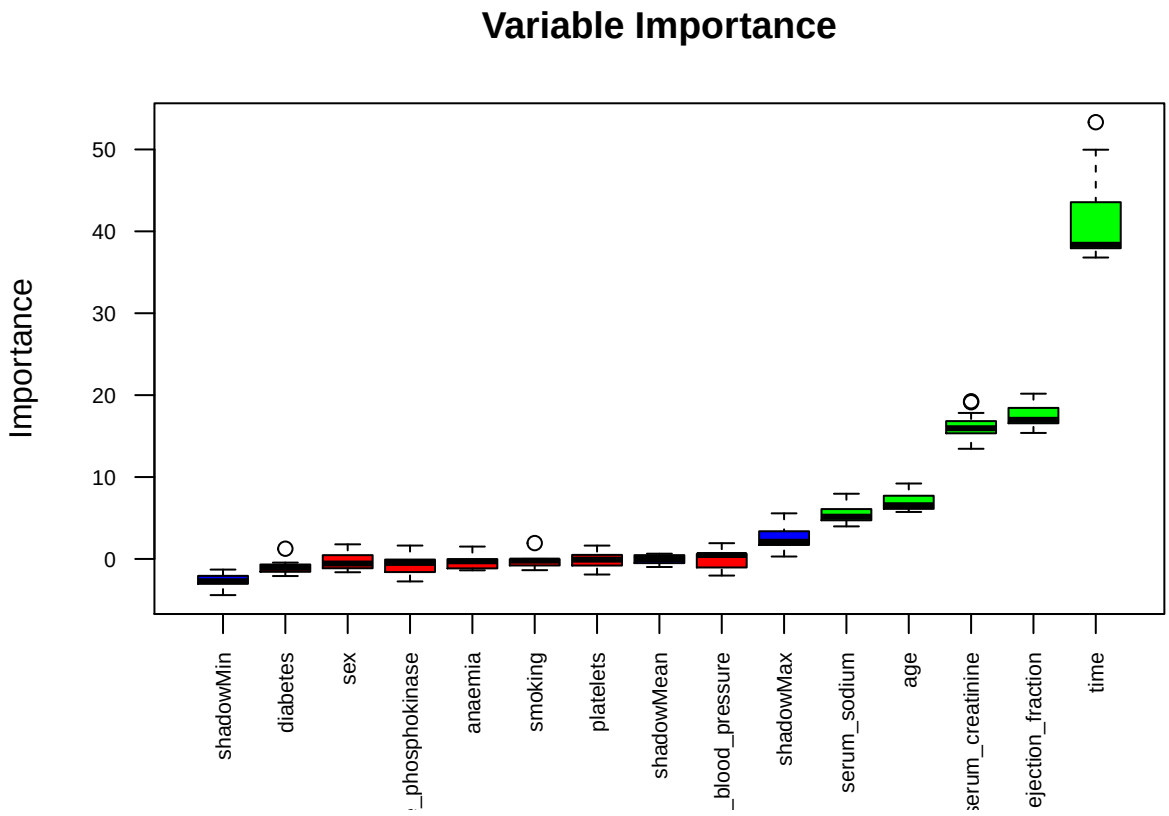
Boruta realiza una búsqueda de tipo Backwards mediante un algoritmo wrapper de Random Forest, y por defecto, usa un p-valor del 0.01 para confirmar cuándo una variable es estadísticamente importante o no.

Vamos a eliminar la variable

```
boruta_output <- Boruta(DEATH_EVENT ~., data = hf, doTrace = 0)

names(boruta_output)
```

```
##  [1] "finalDecision" "ImpHistory"    "pValue"        "maxRuns"
##  [5] "light"         "mcAdj"         "timeTaken"     "roughfixed"
##  [9] "call"          "impSource"
```

```
plot(boruta_output, cex.axis=.7, xlab="", main="Variable Importance", las = 2)
```




```
boruta_signif <- getSelectedAttributes(boruta_output, withTentative = TRUE)
print(boruta_signif)
```

```
## [1] "age"               "ejection_fraction" "serum_creatinine"
## [4] "serum_sodium"      "time"
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.6.3
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
x <- attStats(boruta_output)

ranking <- data.table(attribute=rownames(x), x)[order(-meanImp)]

print(ranking)
```

```
##                    attribute     meanImp   medianImp    minImp    maxImp
##  1:                     time 41.49576225 38.36067427 36.803814 53.327187
##  2:        ejection_fraction 17.39690173 16.98736327 15.383289 20.176891
##  3:         serum_creatinine 16.18179022 15.95641516 13.448651 19.252524
##  4:                      age  6.93540699  6.55692767  5.734912  9.210935
##  5:             serum_sodium  5.45159694  5.15835945  3.974823  7.958874
##  6:                platelets -0.06225754 -0.08419706 -1.903988  1.630401
##  7:      high_blood_pressure -0.08814420  0.48214736 -2.027905  1.924912
##  8:                      sex -0.28515171 -0.56077380 -1.625329  1.782312
##  9:                  smoking -0.30216384 -0.20594389 -1.370779  1.937078
## 10:                  anaemia -0.40382135 -0.28283098 -1.390025  1.511436
## 11: creatinine_phosphokinase -0.61940637 -0.48633609 -2.746921  1.632998
## 12:                 diabetes -0.99972207 -1.08675459 -2.088406  1.250671
##       normHits  decision
##  1: 1.00000000 Confirmed
##  2: 1.00000000 Confirmed
##  3: 1.00000000 Confirmed
##  4: 1.00000000 Confirmed
##  5: 0.93333333 Confirmed
##  6: 0.00000000  Rejected
##  7: 0.00000000  Rejected
##  8: 0.00000000  Rejected
##  9: 0.00000000  Rejected
## 10: 0.00000000  Rejected
## 11: 0.06666667  Rejected
## 12: 0.00000000  Rejected
```