# PRA2 Tipologia de dades

Magí Pàmies Sans

4 de de gener del 2022

## Contents

# 1 Descripció del dataset.

**Perquè és imporant i quina pregunta/problema pretén respondre?**

L'atac de cor és la primera causa de mort de la població adulta, sobretot en paisos desenvolupats. Les seves característiques no el fan un fenomen fàcil de predir tot i que una ràpida intervenció és clau per poder salvar la vida del pacient. Entenent aquesta problemàtica, aquest projecte pretén intentar ajudar a l'equip mèdic a tractar els pacients quan entren a l'hospital amb aquesta patologia.

Per poder resoldre el problema plantejat he seleccionat el joc de dades **Heart failure clinical records Data Set** link. Aquest conté registres de 299 pacients que van patir un atac de cor l'any 2015 (d'abril a desembre) al Pakistan, concretament van ser atesos a l'hospital Allied de Faisalabad. Tots els pacients ja tenien una disfunció sistòlica ventricular esquerra i presentaven insuficències cardíaques previes.

Per a cada registre (pacient), el joc de dades conté 13 variables:

- **age**: age of the patient (years)
- **anaemia**: decrease of red blood cells or hemoglobin (boolean)
- **high blood pressure**: if the patient has hypertension (boolean)
- **creatinine phosphokinase (CPK)**: level of the CPK enzyme in the blood (mcg/L)
- **diabetes**: if the patient has diabetes (boolean)
- **ejection fraction**: percentage of blood leaving the heart at each contraction (percentage)
- **platelets**: platelets in the blood (kiloplatelets/mL)
- **sex**: woman or man (binary)
- **serum creatinine**: level of serum creatinine in the blood (mg/dL)
- **serum sodium**: level of serum sodium in the blood (mEq/L)
- **smoking**: if the patient smokes or not (boolean)
- **time**: follow-up period (days)
- [**target**] **death event**: if the patient deceased during the follow-up period (boolean)

Dins dels objectius que en posem per investigar, el primer serà si hi ha diferències de gènere entre l'edat en que moren els homes i les dones, mitjançant una prova de contrast d'hiòtesis. El segon objectiu serà poder predir si un pacient sobreviurà o no en funció de diverses variables, ja sigui fisològiques, com del històrial mèdic del pacient. Per aquest objectiu farem servir per una banda un model de regressió lineal múltiple i per una altre banda un model d'arbre de decissió no podat.

# 2 Integració i selecció de les dades d' 'interés a analitzar

Carreguem el joc de dades i n'analitzem les variables.

```
# Carreguem el joc de dades
dset <- read.csv('heart_failure_clinical_records_dataset.csv', header = TRUE,
                 sep = ',', fill = F, strip.white = T)

# Verifiquem l'estructura del joc de dades
str(dset)
```

```
## '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 ...
```

Abans de començar amb l'analisis del joc de dades, adaptem el dataset creat amb el fitxer csv. Concretament:

- Canviem el nom de la variable 'creatine_phosphokrinase' per un més curt.
- Convertim en facatorial la variable 'sex', i pasem li posem els valors 'woman' i 'man'.
- convertim a lògic les variables númèriques que són binàries.
- convertim a int la variable age.

Finalment mostrem les primeres files de la taula

```
# Inserim el nom a les columnes
names(dset)[names(dset) == "creatinine_phosphokinase"] <- "creatinine_p"

# Converteixo a int les columnes que els hi toca
dset$age <- as.integer(dset$age)

# Abans de canviar el tipus de valors de les columnes, guardarem una taula per
#poder fer les correlacions amb els valors com a numerics.
dset_2 <- dset

# Converteixo a factorial les columnes que els hi toca
dset$sex <- as.factor(dset$sex)
levels(dset$sex) <- c('woman', 'man')
#levels(dset$sex)[match("0",levels(dset$sex))] <- "woman"
#levels(dset$sex)[match("1",levels(dset$sex))] <- "man"



# Converteixo a lògica les columnes que els hi toca
dset$anaemia <- as.logical(dset$anaemia)
dset$diabetes <- as.logical(dset$diabetes)
dset$high_blood_pressure <- as.logical(dset$high_blood_pressure)
dset$smoking <- as.logical(dset$smoking)
dset$DEATH_EVENT <- as.logical(dset$DEATH_EVENT)

# Mostrem les primers files
head(dset)
```

```
##   age anaemia creatinine_p diabetes ejection_fraction high_blood_pressure
## 1  75   FALSE          582    FALSE                20                TRUE
## 2  55   FALSE         7861    FALSE                38               FALSE
## 3  65   FALSE          146    FALSE                20               FALSE
## 4  50    TRUE          111    FALSE                20               FALSE
## 5  65    TRUE          160     TRUE                20               FALSE
## 6  90    TRUE           47    FALSE                40                TRUE
##   platelets serum_creatinine serum_sodium   sex smoking time DEATH_EVENT
```

```
## 1    265000              1.9          130   man   FALSE    4        TRUE
## 2    263358              1.1          136   man   FALSE    6        TRUE
## 3    162000              1.3          129   man    TRUE    7        TRUE
## 4    210000              1.9          137   man   FALSE    7        TRUE
## 5    327000              2.7          116 woman   FALSE    8        TRUE
## 6    204000              2.1          132   man    TRUE    8        TRUE
```

```
# Verifiquem l'estructura del joc de dades
str(dset)
```

```
## 'data.frame':    299 obs. of  13 variables:
##  $ age               : int  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia           : logi  FALSE FALSE FALSE TRUE TRUE TRUE ...
##  $ creatinine_p      : int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes          : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ ejection_fraction : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure: logi  TRUE FALSE FALSE FALSE FALSE TRUE ...
##  $ 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               : Factor w/ 2 levels "woman","man": 2 2 2 2 1 2 2 2 1 2 ...
##  $ smoking           : logi  FALSE FALSE TRUE FALSE FALSE TRUE ...
##  $ time              : int  4 6 7 7 8 8 10 10 10 10 ...
##  $ DEATH_EVENT       : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
```

Podem observa que la taula té 13 variables i 299 observacions. Les variables(columnes) que té són:

- **age** int, anys del pacient (anys).
- **anemia** logic, disminució de globus vermells o hemoglobina.
- **high_blood_presure** logic, si el pacient té hipertenció.
- **creatinine_p** int, nivell de l'enzim CPK a la sang (mcg/L).
- **biabetes** logic, si el pacient té diabetis.
- **ejection_fraction** int, percentatge de sang que emet el cor en cada contracció (percentatge).
- **platelets** num, concentració de paletes a la sang (kiloplatelets/mL).
- **serum_creatinine** num, nivell de serum creatinine a la sang (mg/dL).
- **serum_sodium** int, nivell de serum sodium a la sang (mEq/dL).
- **sex** factor, génere del pacient (Home/Dona).
- **smoking** logic, si el pacient fuma.
- **time** int, periode de seguiment (dies).
- **DEATH_EVENT** logic, si el pacient ha mort durant el periode de seguiment.

# 3 Neteja de les dades.

## 3.1 Les dades contenen zeros o elements buits? Com gestionaries aquests casos?

Analizem si les dades contenen zeros o elements buits.

```
# Analisis de valors buits
# colSums(is.na(dset))
# colSums(dset=="")
# colSums(dset=="?")
#nas2 <- sapply(dset, function(x) sum(dset==""))
#nas3 <- sapply(dset, function(x) sum(dset=="?"))

# Analisis de valors buits
nas <- sapply(dset, function(x) sum(is.na(x)))
kable(data.frame(Variables = names(nas), NAs = as.vector(nas)))
```

| Variables | NAs |
|---|---|
| age | 0 |
| anaemia | 0 |
| creatinine_p | 0 |
| diabetes | 0 |
| ejection_fraction | 0 |
| high_blood_pressure | 0 |
| platelets | 0 |
| serum_creatinine | 0 |
| serum_sodium | 0 |
| sex | 0 |
| smoking | 0 |
| time | 0 |
| DEATH_EVENT | 0 |

De l'anàlisis dels valors Nodata i dels valors buits, podem observar que no en tenim. Les variables numériques no tenen cap valor nodata ni buit.

## 3.2 Identificació i tractament de valors extrems.

```
# Analitzem els valors extrems
outliers <- sapply(dset, function(x) paste(boxplot.stats(x)$out,collapse=" "))
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors
```

```
kable(data.frame(variables=names(outliers),clase=as.vector(outliers)))
```

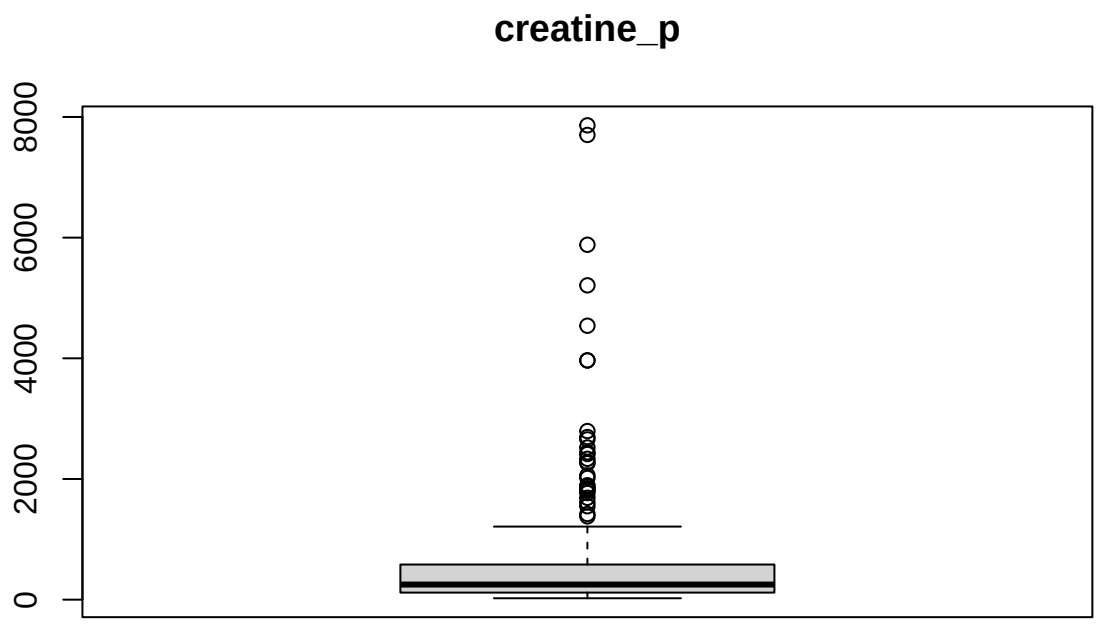
| variables | clase |
|---|---|
| age | |
| anaemia | |
| creatinine_p | 7861 2656 1380 3964 7702 5882 5209 1876 1808 4540 1548 1610 2261 1846 2334 2442 3966 1419 1896 1767 2281 2794 2017 2522 2695 1688 1820 2060 2413 |
| diabetes | |
| ejection_fraction | 80 70 |
| high_blood_pressure | |

| variables | clase |
|---|---|
| platelets | 454000 47000 451000 461000 497000 621000 850000 507000 448000 75000 70000 73000 481000 504000 62000 533000 25100 451000 51000 543000 742000 |
| serum_creatinine | 2.7 9.4 4 5.8 3 3.5 2.3 3 4.4 6.8 2.2 2.7 2.3 2.9 2.5 2.3 3.2 3.7 3.4 6.1 2.5 2.4 2.5 3.5 9 5 2.4 2.7 3.8 |
| serum_sodium | 116 121 124 113 |
| sex | |
| smoking | |
| time | |
| DEATH_EVENT | |

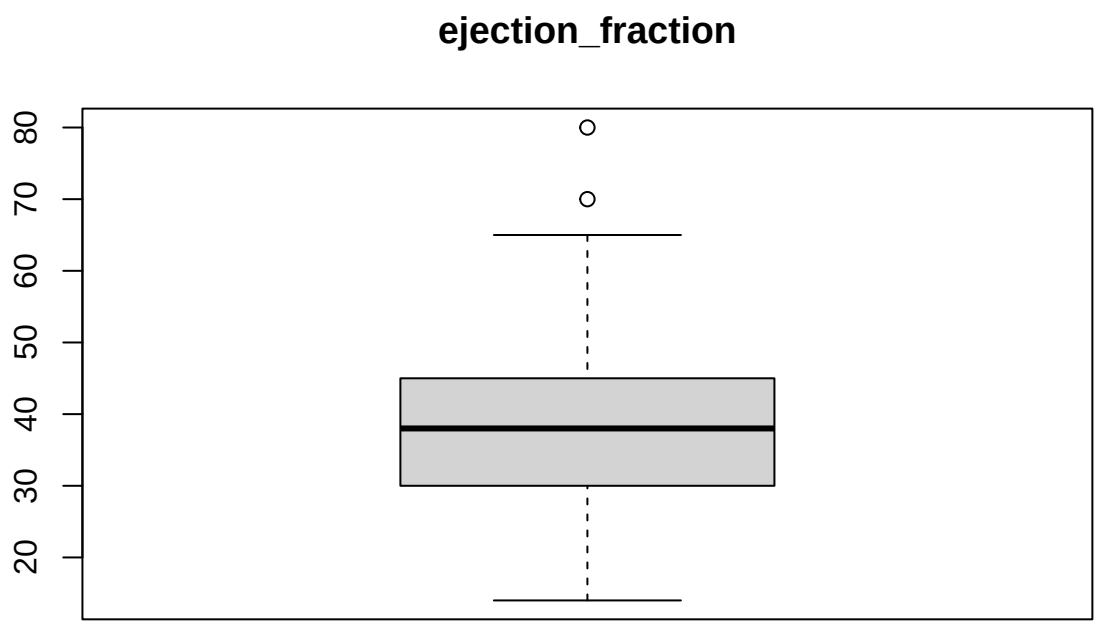
```
# Estadístiques bàsiques del dataset
summary(dset)
```

```
##       age         anaemia         creatinine_p     diabetes
##  Min.   :40.00   Mode :logical   Min.   :  23.0   Mode :logical
##  1st Qu.:51.00   FALSE:170       1st Qu.: 116.5   FALSE:174
##  Median :60.00   TRUE :129       Median : 250.0   TRUE :125
##  Mean   :60.83                   Mean   : 581.8
##  3rd Qu.:70.00                   3rd Qu.: 582.0
##  Max.   :95.00                   Max.   :7861.0
##  ejection_fraction high_blood_pressure   platelets      serum_creatinine
##  Min.   :14.00     Mode :logical       Min.   : 25100   Min.   :0.500
##  1st Qu.:30.00     FALSE:194           1st Qu.:212500   1st Qu.:0.900
##  Median :38.00     TRUE :105           Median :262000   Median :1.100
##  Mean   :38.08                         Mean   :263358   Mean   :1.394
##  3rd Qu.:45.00                         3rd Qu.:303500   3rd Qu.:1.400
##  Max.   :80.00                         Max.   :850000   Max.   :9.400
##   serum_sodium      sex       smoking             time       DEATH_EVENT
##  Min.   :113.0   woman:105   Mode :logical   Min.   :  4.0   Mode :logical
##  1st Qu.:134.0   man  :194   FALSE:203       1st Qu.: 73.0   FALSE:203
##  Median :137.0               TRUE :96        Median :115.0   TRUE :96
##  Mean   :136.6                               Mean   :130.3
##  3rd Qu.:140.0                               3rd Qu.:203.0
##  Max.   :148.0                               Max.   :285.0
```

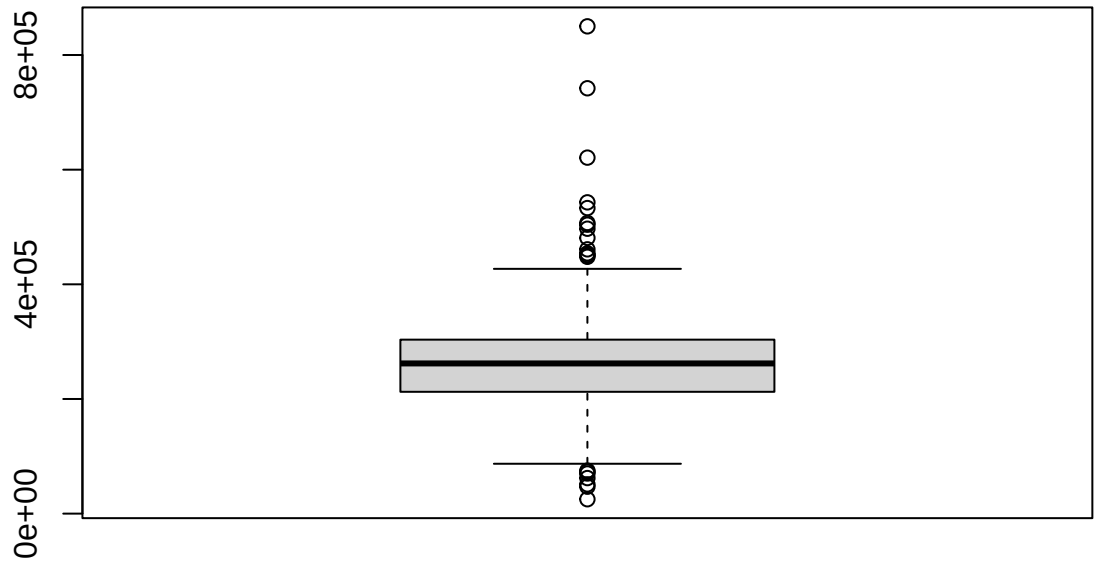
```
# Mostrem les variables que tenen valors extrems
boxplot(dset$creatinine_p, main="creatine_p")
```



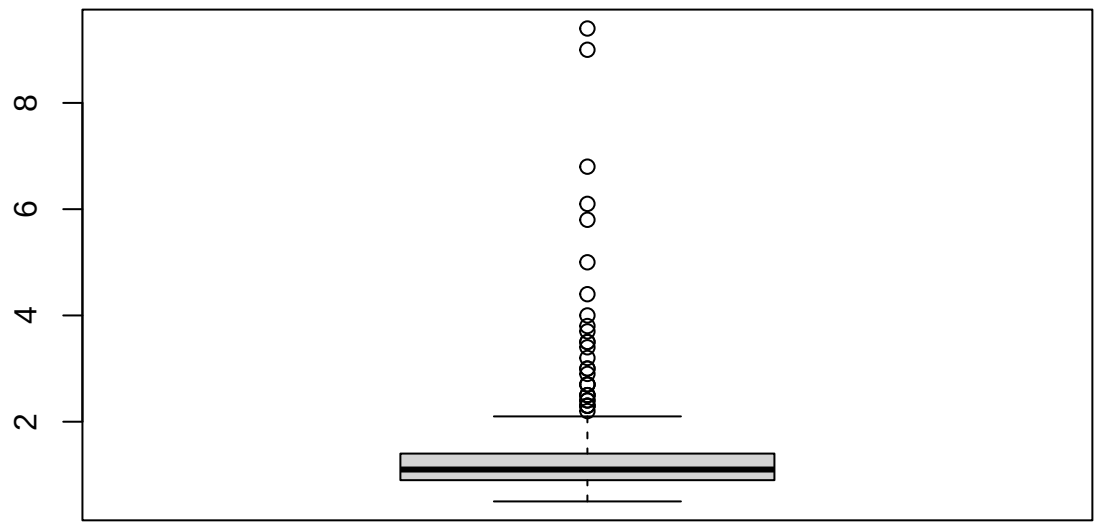

```
boxplot(dset$ejection_fraction, main="ejection_fraction")
```



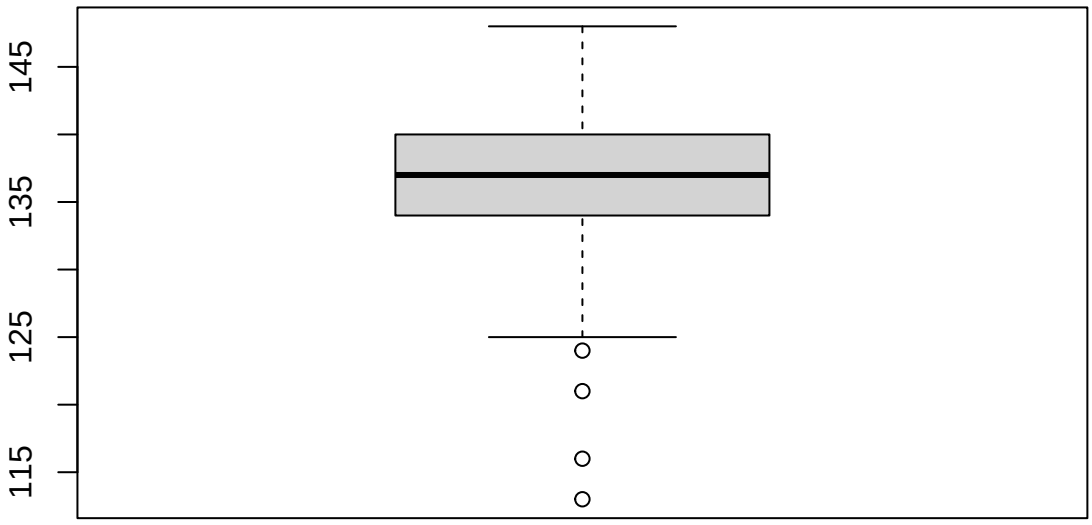

```
boxplot(dset$platelets, main="platelets")
```




```
boxplot(dset$serum_creatinine, main="serum_creatinine")
```




```
boxplot(dset$serum_sodium, main="serum_sodium")
```




Observem que les variables “creatine_p”“ejection_fraction”, “platelets”, “serum_creatinine” i “serum_sodium” tenen valors extrems. Analitzant variable per variable observem que aquests valors estan dins del rang que aquestes variables poden acceptar, per tant considerem que és millor deixar-los en l’estudi, ja que ens poden explicar cosa i no tindria cap sentit treure’ls.

# 4 Anàlisis de les dades.

## 4.1 Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar).

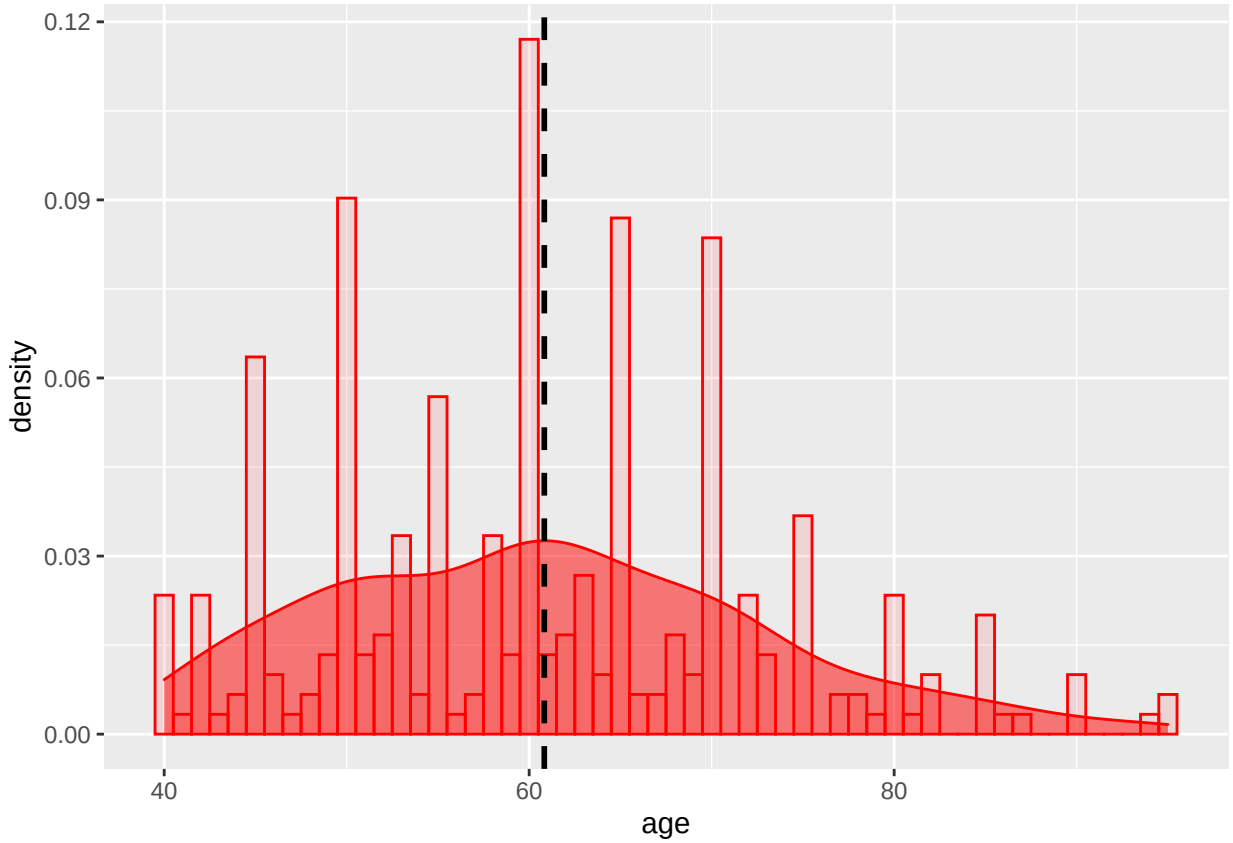
Una de les coses que podem fer abans de començar a analitzar les dades és crear o modificar algunes variables per fer-nos més senzill l'anàlisis. En aquest sentit, podem categoritzar algunes de les variables continues, per tal de tenir més valors a introduir en l'arbre de decisió.

Podem crear una altre variable que ens categoritzi la gent en funció de la seva edad, agrupant-los per grups d'edat. Comencem categoritzan la variable age, primer analitzem com es distribueix:

```
summary(dset[,"age"])
```

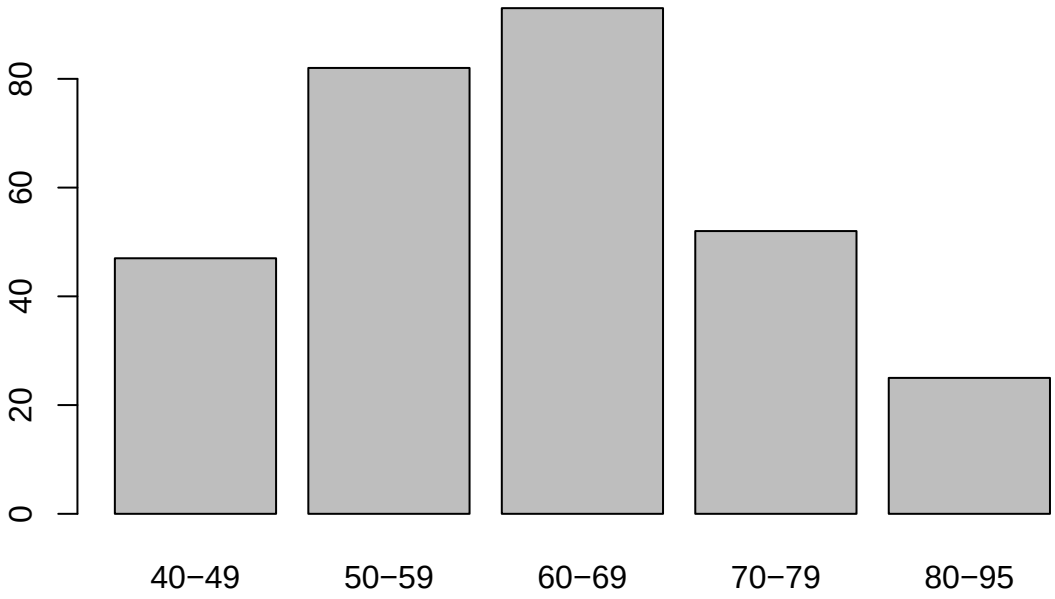
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40.00   51.00   60.00   60.83   70.00   95.00
```

```
# Observem com es distribueixen
ggplot(dset, aes(age))+
  geom_histogram(col='red', fill='red', alpha = 0.1, binwidth = 1,
                 aes(y = ..density..))+
  geom_density(col='red', fill='red', alpha=0.5)+
  geom_vline(aes(xintercept=mean(age)),
             color=1, linetype='dashed', size=1)
```




Observem que els les edats amb major freqüencia s'agrupen entre els 50 i els 70 anys, la mitjan a està al voltant dels 60 anys, l'edat mínima és de 40 anys i la màxima de 95. Observem que a partir dels 80 anys el número de pacients és molt baix. Podem agrupar els pacients en grups de 10 anys, menys l'últim grup que serà de 15, ja que el número de pacients d'entre 80 i 95 és molt baix.

```r
# Creem la nova variable
dset["segment_age"] <- cut(dset$age, breaks = c(40,50,60,70,80,96),
                           labels = c("40-49", "50-59", "60-69","70-79","80-95"),
                           right = FALSE)
# Mostrem una gràfica per veure com es distribueixen aquestes categories noves
plot(dset$segment_age)
```

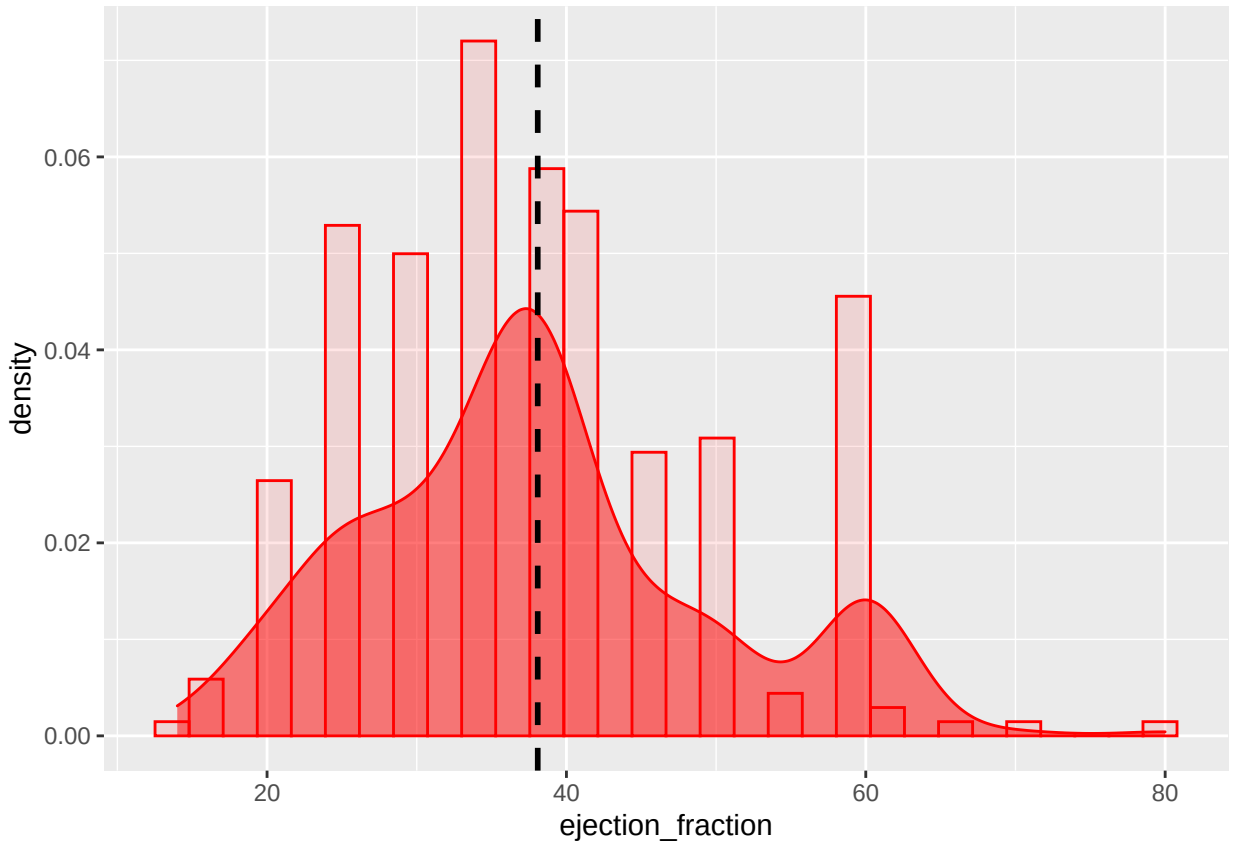



Una altre categoria que podem categoritzar és la 'ejection fraction'. Analitzem com es distribueix.

```r
summary(dset[,"ejection_fraction"])
```

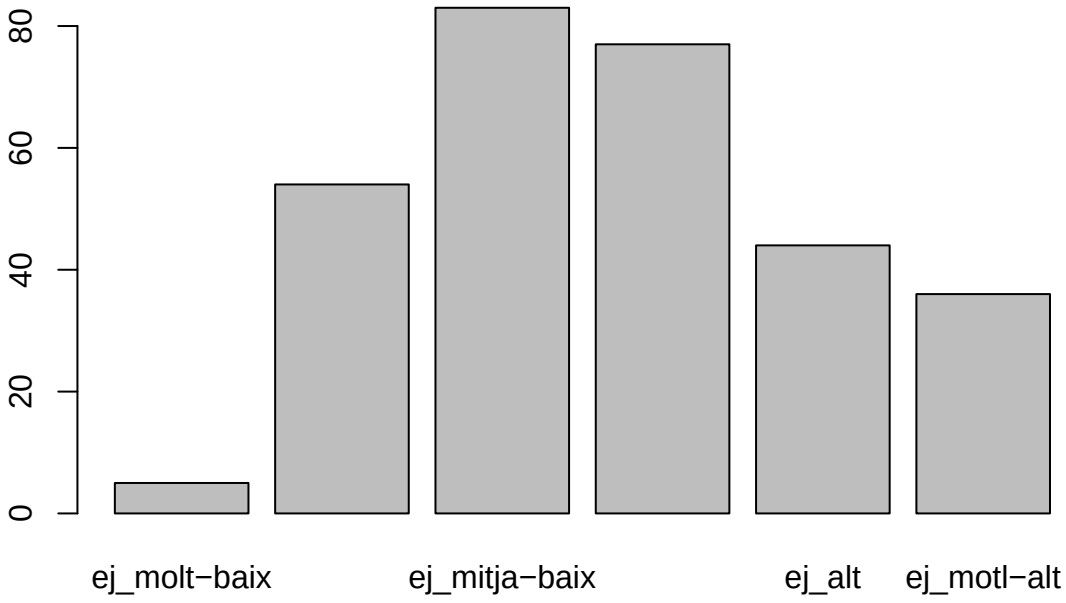
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.00   30.00   38.00   38.08   45.00   80.00
```

```r
# Observem com es distribueixen
ggplot(dset, aes(ejection_fraction))+
  geom_histogram(col='red', fill='red', alpha = 0.1, aes(y = ..density..))+
  geom_density(col='red', fill='red', alpha=0.5)+
  geom_vline(aes(xintercept=mean(ejection_fraction)),
             color=1, linetype='dashed', size=1)
```




Observem que els valors fluctuen entre el 14 i el 80, la mitjana està en 38 i la majoria de valors s'agrupen entre els 30 i els 45. Tinguén un lleuger repunt entre el 55 i el 65. A partir del 65 el numero de registres és molt petit. A l'hora de categoritzar aquesta variable, i tenint en compte que estem parlant d'uns valors que representen un percentatge, ho podem categoritzar entre molt-baix (inferior a 20) baix (entre 20 i el primer quartil), mitja-baix (entre el primer i el segon quartil), mitja-alt(entre el segon i tercer quartil), alt(entre el tercer quartil i 60) i molt-alt(entre 60 i 80).

```
# Creem la nova variable
dset["segment_ejection"] <- cut(dset$ejection_fraction,
                                breaks = c(14,20,30,38,45,60,81),
                                labels = c("ej_molt-baix", "ej_baix",
                                "ej_mitja-baix", "ej_mitja-alt","ej_alt",
                                "ej_motl-alt"), right = FALSE)
# Mostrem una gràfica per veure com es distribueixen aquestes categories noves
plot(dset$"segment_ejection")
```

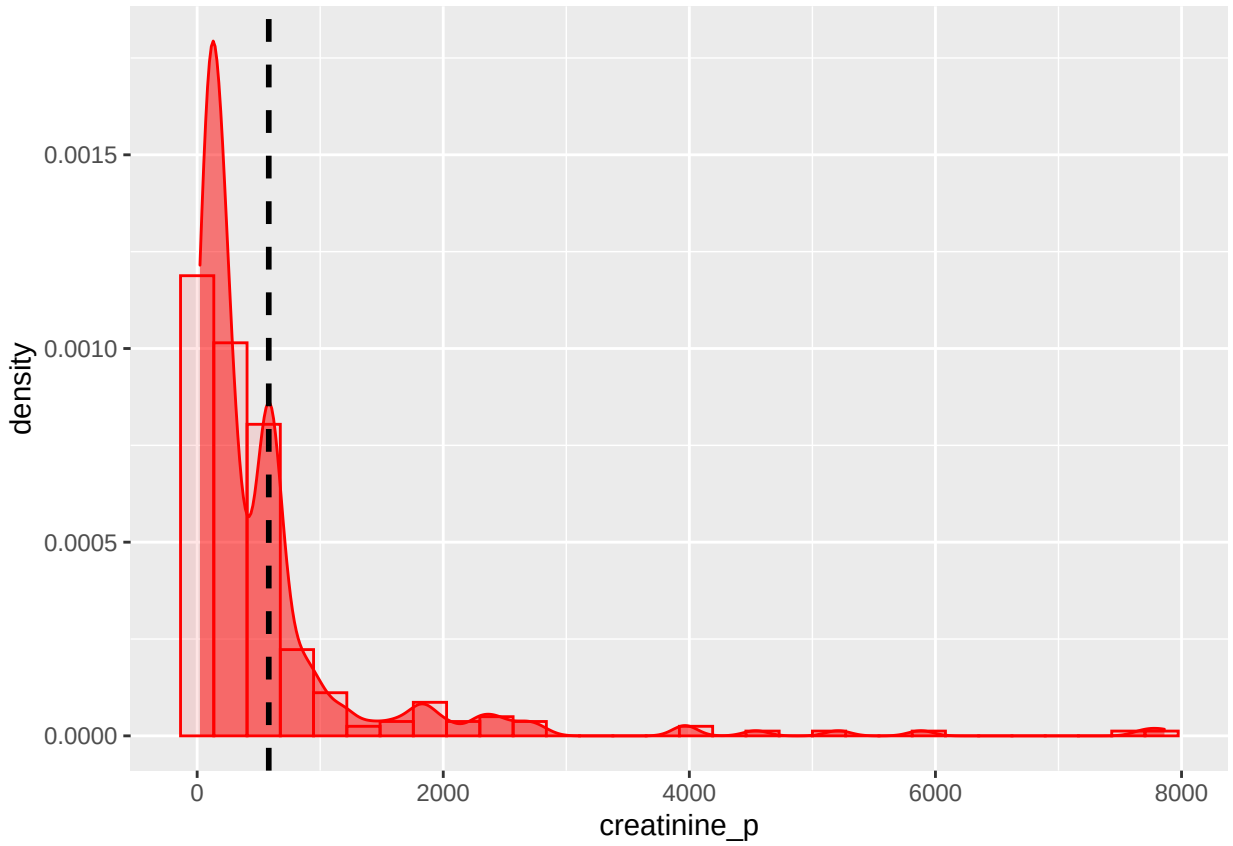



Una altre categoria que podem categoritzar és la 'creatinine_p'. Analitzem com es distribueix.

```
summary(dset[,"creatinine_p"])
```

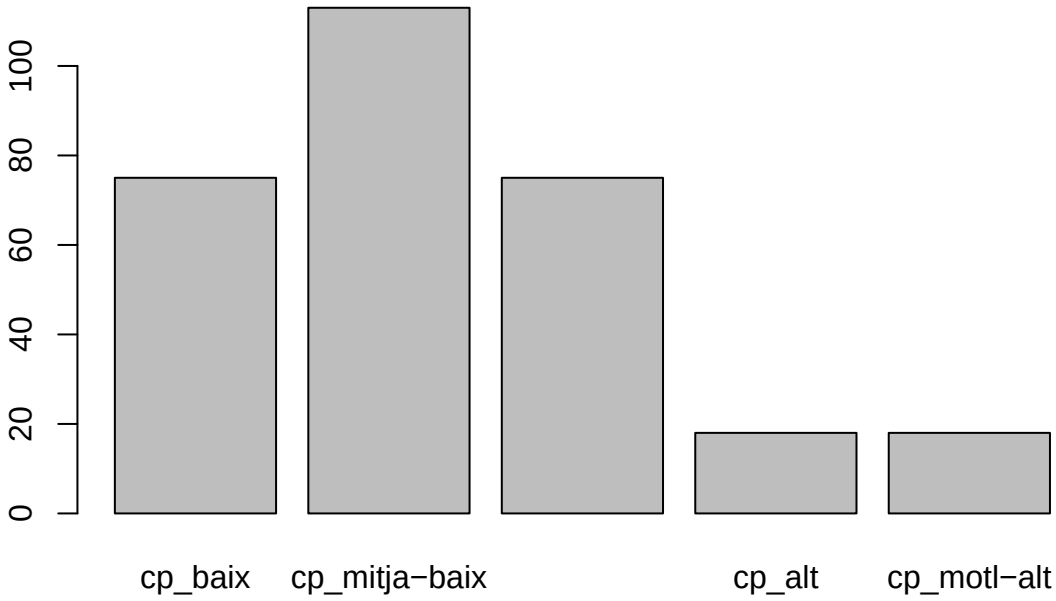
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    23.0   116.5   250.0   581.8   582.0  7861.0
```

```
# Observem com es distribueixen
ggplot(dset, aes(creatinine_p))+
  geom_histogram(col='red', fill='red', alpha = 0.1, aes(y = ..density..))+
  geom_density(col='red', fill='red', alpha=0.5)+
  geom_vline(aes(xintercept=mean(creatinine_p)),
             color=1, linetype='dashed', size=1)
```




En aquest cas observem que els valors es concentren el 100 i el 600, però després trobem registres fins els 8000. Una opció és agar els quartils per agrupar els valors en quatre grups i després crear-ne dos més que agrupin els valors que més s'allunyen de la mitjana per la part superior de l'eix de les x.

```
# Creem la nova variable
dset["segment_creatine_p"] <- cut(dset$creatinine_p, breaks = c(23,116,581,1000,
                                  2000,7862), labels = c("cp_baix", "cp_mitja-baix",
                                  "cp_mitja-alt","cp_alt", "cp_motl-alt"),
                                  right = FALSE)
# Mostrem una gràfica per veure com es distribueixen aquestes categories noves
plot(dset$"segment_creatine_p")
```

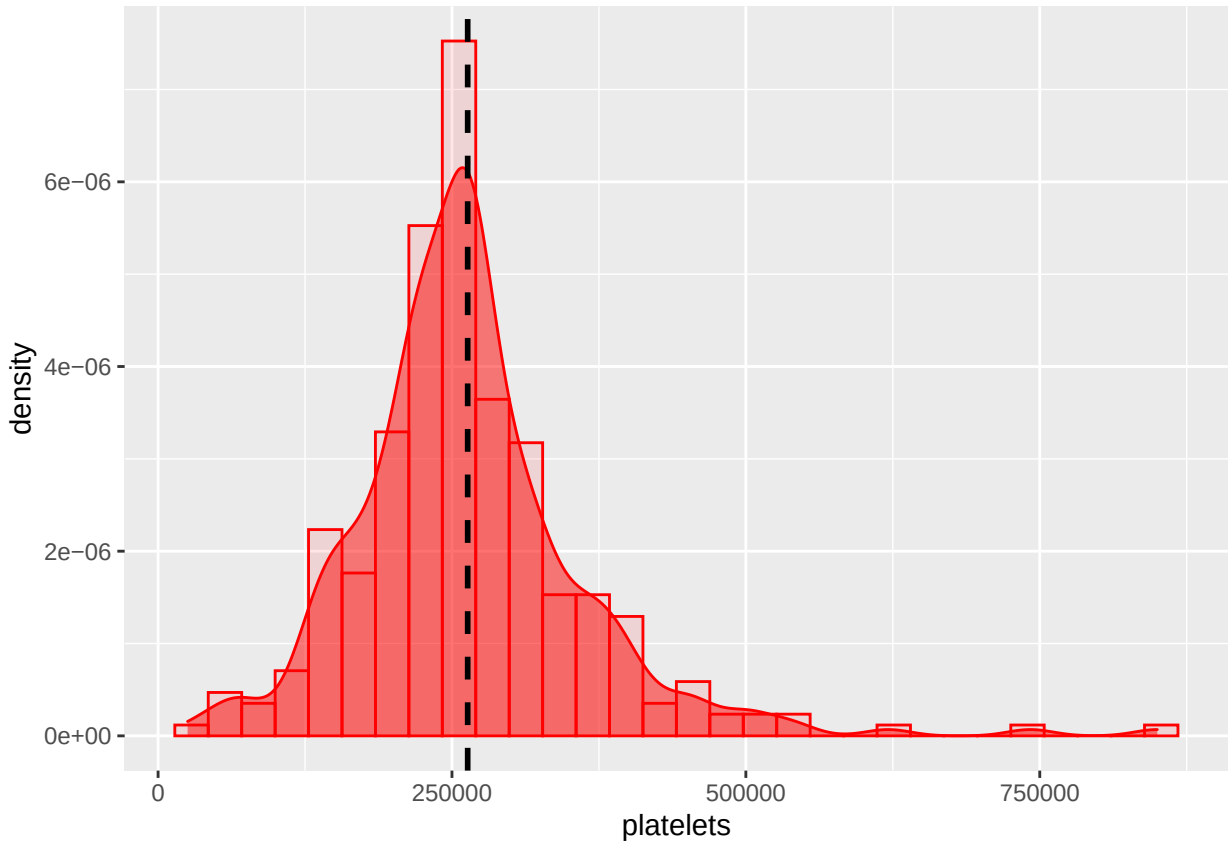



Una altre categoria que podem categoritzar és la 'platelets'. Analitzem com es distribueix.

```
summary(dset[,"platelets"])
```

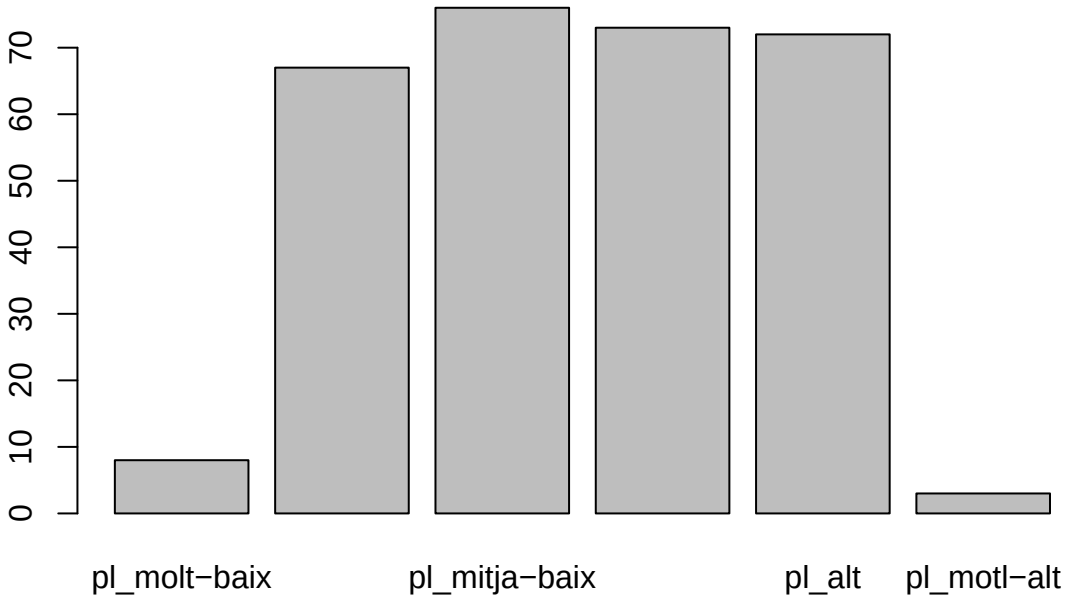
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25100  212500  262000  263358  303500  850000
```

```
# Observem com es distribueixen
ggplot(dset, aes(platelets))+
  geom_histogram(col='red', fill='red', alpha = 0.1, aes(y = ..density..))+
  geom_density(col='red', fill='red', alpha=0.5)+
  geom_vline(aes(xintercept=mean(platelets)),
             color=1, linetype='dashed', size=1)
```




Observem que la distribució dels valors s'aproxima a una distribució normal, tot i que s'allarga el final per la part superior. És a dir que des del tercer quartil fins a l'últim valor la distància és molt gran tot i que el número de registres és molt baix. Agruparem els valors en 6 categories, en que la primera i l'útlima agafaran els valors més allunyats de la mitjana i que tot i tenir un rang molt ampli representen un percentatge molt baix dels valors. Les altres quatre categories agafaran els valors més centrals.

```
# Creem la nova variable
dset["segment_platelets"] <- cut(dset$platelets, breaks = c(25100,100000,212500,
                                 263358,303500,600000,850001), labels = c("pl_molt-baix",
                                 "pl_baix", "pl_mitja-baix", "pl_mitja-alt",
                                 "pl_alt", "pl_motl-alt"), right = FALSE)
# Mostrem una gràfica per veure com es distribueixen aquestes categories noves
plot(dset$"segment_platelets")
```

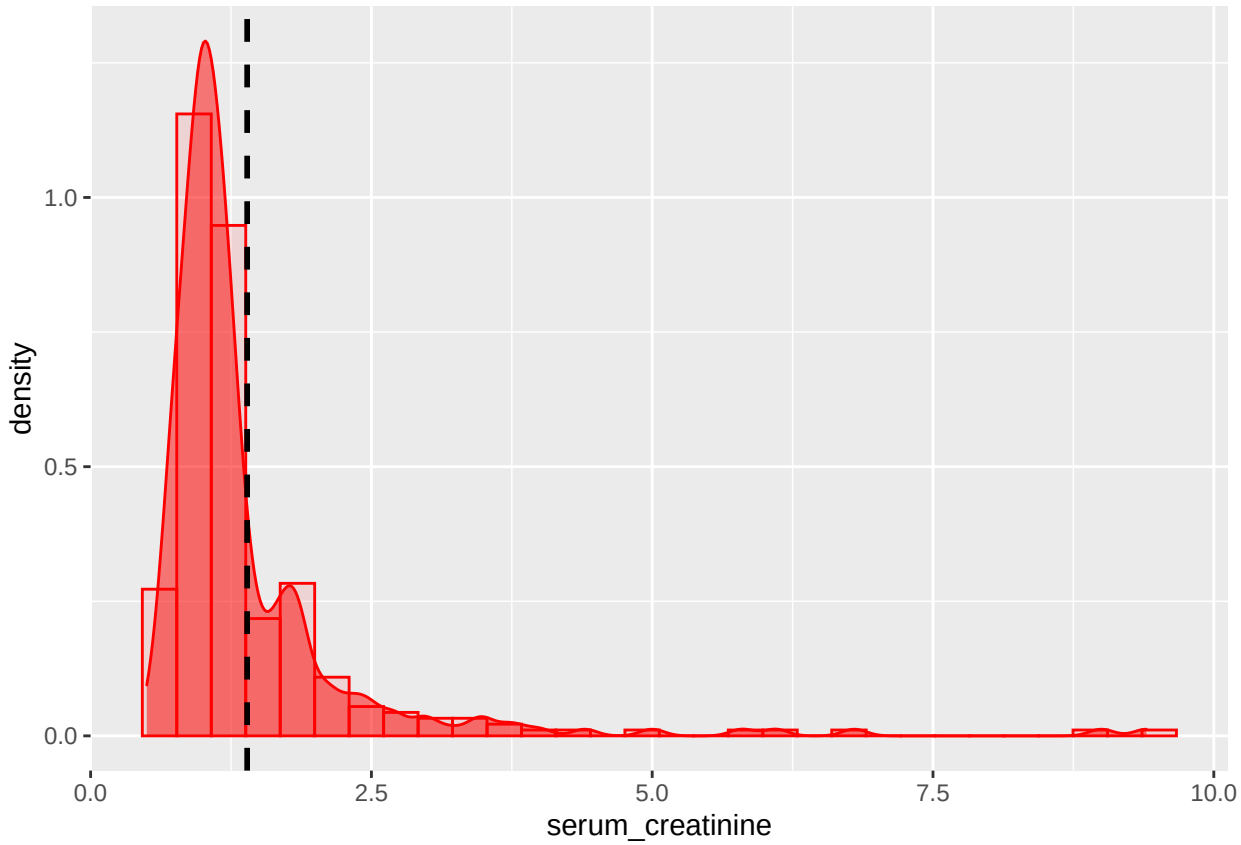



Una altre categoria que podem categoritzar és la 'serum_creatinine'. Analitzem com es distribueix.

```
summary(dset[,"serum_creatinine"])
```

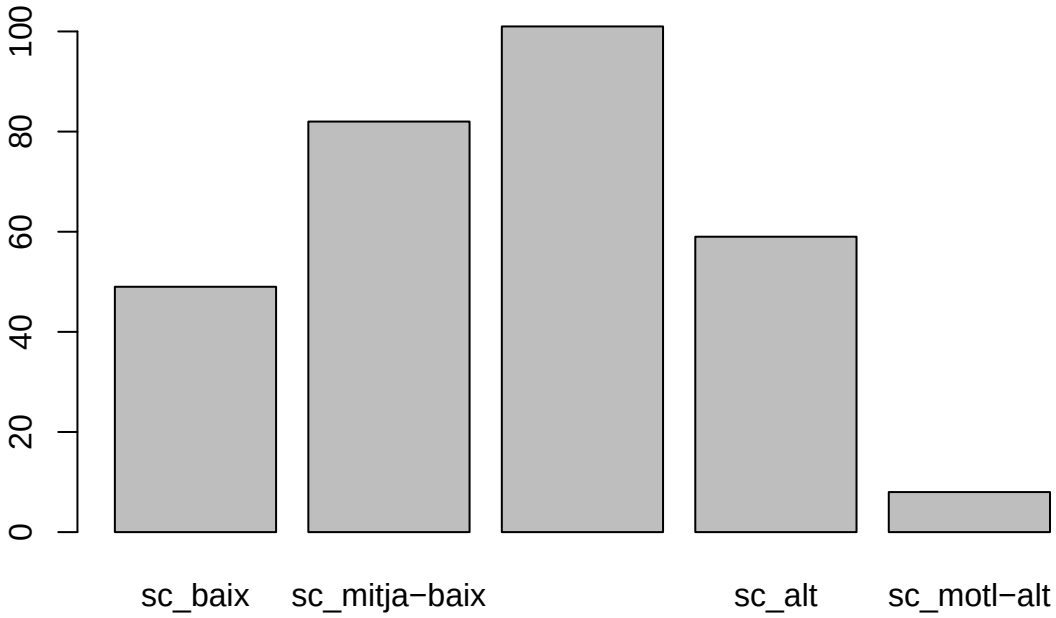
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   0.900   1.100   1.394   1.400   9.400
```

```
# Observem com es distribueixen
ggplot(dset, aes(serum_creatinine))+
  geom_histogram(col='red', fill='red', alpha = 0.1, aes(y = ..density..))+
  geom_density(col='red', fill='red', alpha=0.5)+
  geom_vline(aes(xintercept=mean(serum_creatinine)),
             color=1, linetype='dashed', size=1)
```




Ens trobem amb una situació similiar a la variable 'creatinine_p', en que és valors es concentren en un rang reduit i després la gràfica s'allarga en l'eix de les x per la part superior però amb un número de registres molt reduit. Agruparem els valors en 5 categories, en que la útlima agafaran els valors més allunyats de la mitjana i que tot i tenir un rang molt ampli representen un percentatge molt baix dels valors. Les altres quatre categories agafaran els valors més centrals.

```
# Creem la nova variable
dset["segment_serum_creatinine"] <- cut(dset$serum_creatinine, breaks = c(0.500,
                                        0.900,1.100,1.600,4.000,9.401), labels =
                                        c("sc_baix", "sc_mitja-baix", "sc_mitja-alt",
                                        "sc_alt", "sc_motl-alt"), right = FALSE)
# Mostrem una gràfica per veure com es distribueixen aquestes categories noves
plot(dset$"segment_serum_creatinine")
```

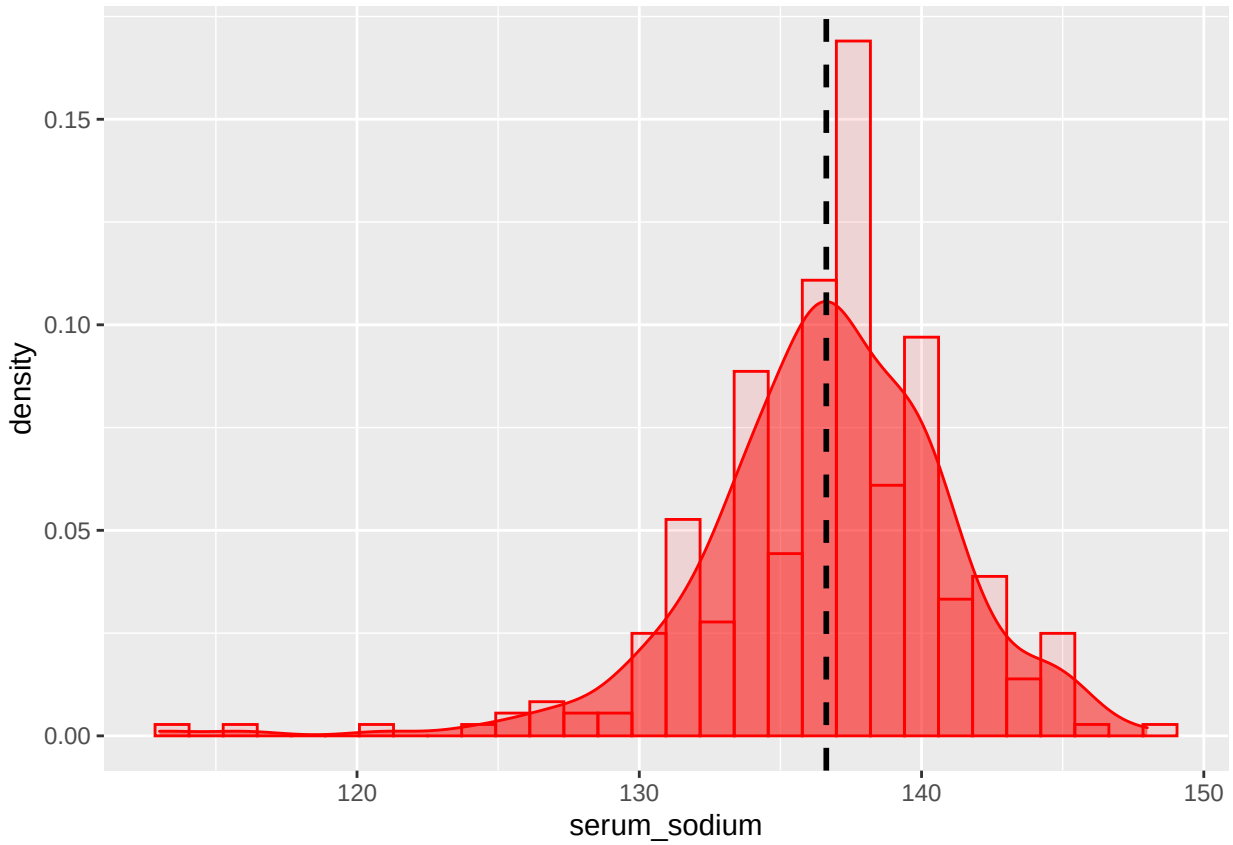



Una altre categoria que podem categoritzar és la 'serum_sodium'. Analitzem com es distribueix.

```
summary(dset[,"serum_sodium"])
```

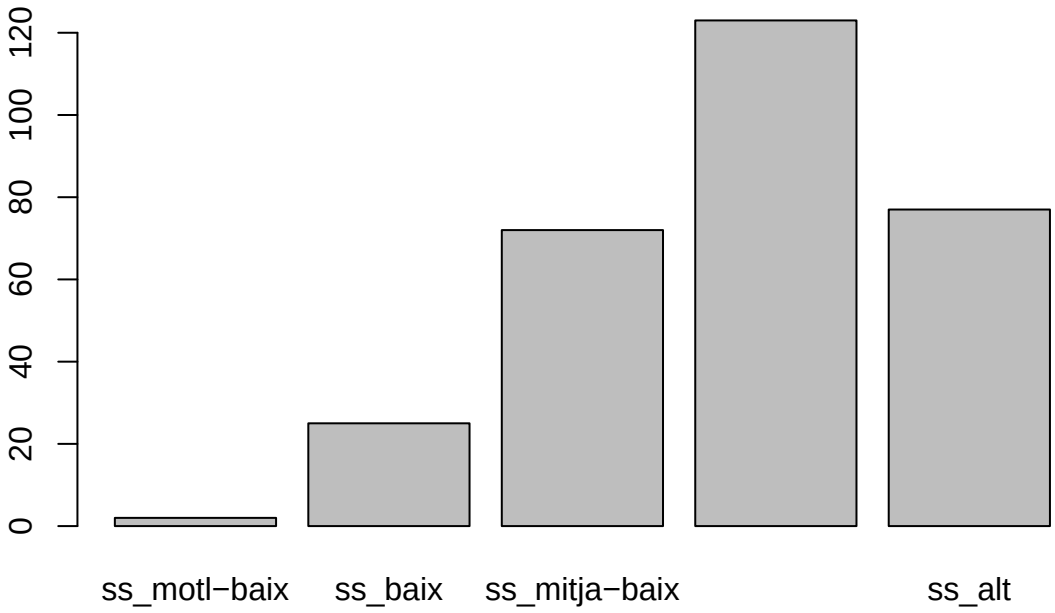
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   113.0   134.0   137.0   136.6   140.0   148.0
```

```
# Observem com es distribueixen
ggplot(dset, aes(serum_sodium))+
  geom_histogram(col='red', fill='red', alpha = 0.1, aes(y = ..density..))+
  geom_density(col='red', fill='red', alpha=0.5)+
  geom_vline(aes(xintercept=mean(serum_sodium)),
             color=1, linetype='dashed', size=1)
```




Ens trobem amb una situació similiar a la variable 'platelets' però de forma inversa. És a dir que els valors és concentre a la part final de l'eix de les x. Mentres a que a la part inicial hi trobem un rang de nombres molt elevat però un número de regístres molt baix. Agruparem els valors en 5 categories, en que la primera agafaran els valors més allunyats de la mitjana a l'inici de l'eix de les x, que tot i tenir un rang molt ampli representen un percentatge molt baix dels valors. Les altres quatre categories agafaran els valors més centrals.

```
# Creem la nova variable
dset["segment_serum_sodium"] <- cut(dset$serum_sodium, breaks = c(113,120,132,
                                    136,140,149), labels = c("ss_motl-baix","ss_baix",
                                    "ss_mitja-baix", "ss_mitja-alt","ss_alt"), right = FALSE)
# Mostrem una gràfica per veure com es distribueixen aquestes categories noves
plot(dset$"segment_serum_sodium")
```




Analitzem com ens ha quedat el dataset.

```
summary(dset)
```

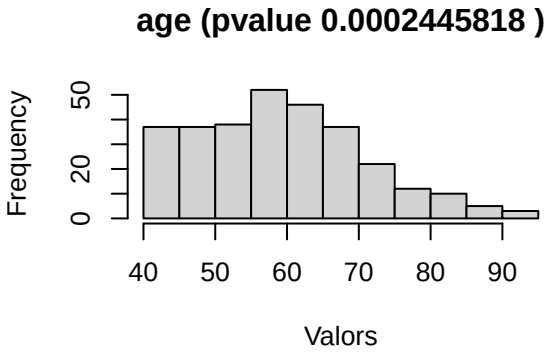
```
##       age         anaemia         creatinine_p     diabetes
##  Min.   :40.00   Mode :logical   Min.   :  23.0   Mode :logical
##  1st Qu.:51.00   FALSE:170       1st Qu.: 116.5   FALSE:174
##  Median :60.00   TRUE :129       Median : 250.0   TRUE :125
##  Mean   :60.83                   Mean   : 581.8
##  3rd Qu.:70.00                   3rd Qu.: 582.0
##  Max.   :95.00                   Max.   :7861.0
##  ejection_fraction high_blood_pressure   platelets      serum_creatinine
##  Min.   :14.00     Mode :logical       Min.   : 25100   Min.   :0.500
##  1st Qu.:30.00     FALSE:194           1st Qu.:212500   1st Qu.:0.900
##  Median :38.00     TRUE :105           Median :262000   Median :1.100
##  Mean   :38.08                         Mean   :263358   Mean   :1.394
##  3rd Qu.:45.00                         3rd Qu.:303500   3rd Qu.:1.400
##  Max.   :80.00                         Max.   :850000   Max.   :9.400
##   serum_sodium      sex       smoking            time       DEATH_EVENT
##  Min.   :113.0   woman:105   Mode :logical   Min.   :  4.0   Mode :logical
##  1st Qu.:134.0   man  :194   FALSE:203       1st Qu.: 73.0   FALSE:203
##  Median :137.0               TRUE :96        Median :115.0   TRUE :96
##  Mean   :136.6                               Mean   :130.3
##  3rd Qu.:140.0                               3rd Qu.:203.0
##  Max.   :148.0                               Max.   :285.0
##  segment_age      segment_ejection     segment_creatine_p     segment_platelets
##  40-49:47    ej_molt-baix : 5      cp_baix      : 75      pl_molt-baix : 8
##  50-59:82    ej_baix      :54      cp_mitja-baix:113      pl_baix      :67
##  60-69:93    ej_mitja-baix:83      cp_mitja-alt : 75      pl_mitja-baix:76
##  70-79:52    ej_mitja-alt :77      cp_alt       : 18      pl_mitja-alt :73
##  80-95:25    ej_alt       :44      cp_motl-alt  : 18      pl_alt       :72
##              ej_motl-alt  :36                             pl_motl-alt  : 3
##   segment_serum_creatinine    segment_serum_sodium
##  sc_baix      : 49         ss_motl-baix :  2
##  sc_mitja-baix: 82         ss_baix      : 25
##  sc_mitja-alt :101         ss_mitja-baix: 72
```

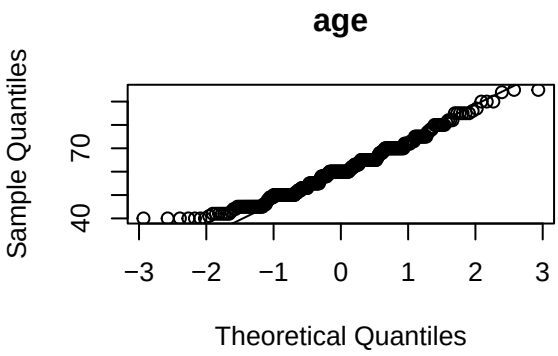
```
##  sc_alt       : 59         ss_mitja-alt :123
##  sc_motl-alt  :  8         ss_alt       : 77
##
```

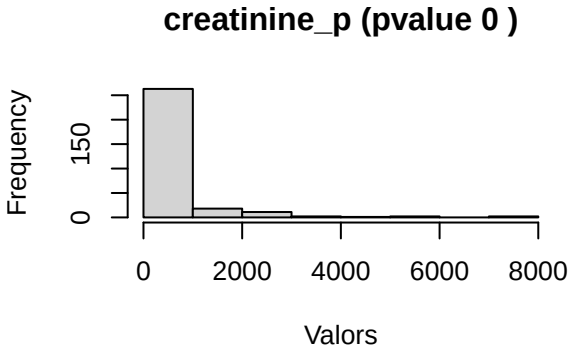
## 4.2 Comprovació de la normalitat i homogeneïtat de la variància.

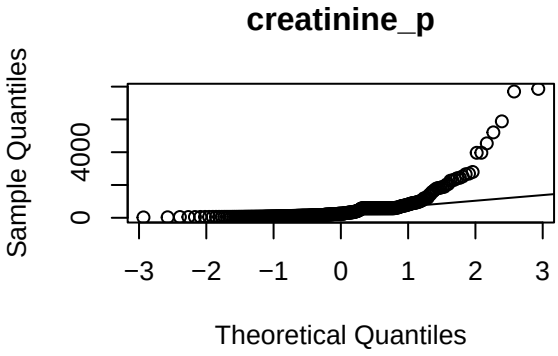
Comprovem si les variables quantitatives tenen una distribució normal.

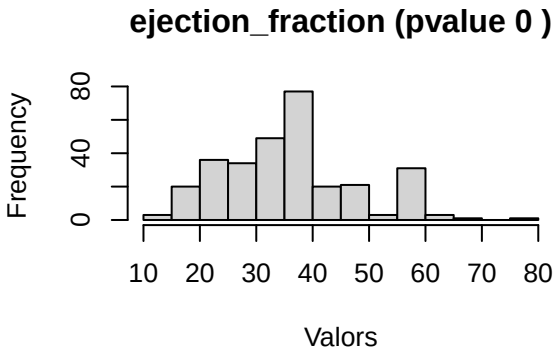
```
# Perquè ens aparelli les gràfiques
par(mfrow = c(2, 2))

# Li diem que de les variables numériques, ens mostri la gràfica de les
#que tenen un pvalue inferior a 0.05
for (i in colnames(dset)) {
  if (class(dset[,i]) !='logical' & !is.factor(dset[,i])){
    if(lillie.test(dset[,i])$p.value < 0.05){
      hist(dset[,i], main = paste(i, "(pvalue",round(lillie.test(dset[,i])$p.value,10),
                                  ")"), xlab = 'Valors')
    qqnorm(dset[,i], main = i)
    qqline(dset[,i])
    }

  }

}
```

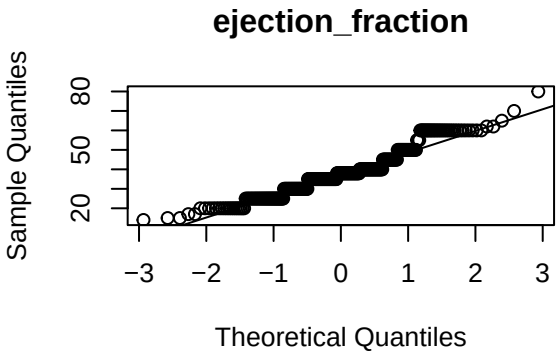


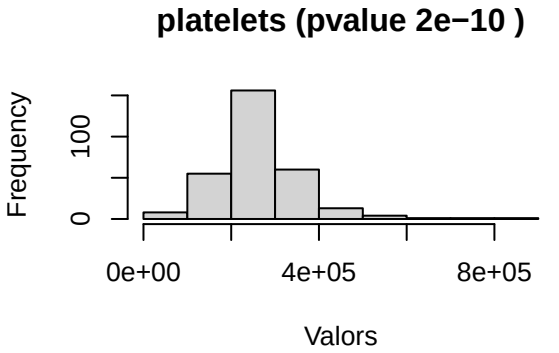



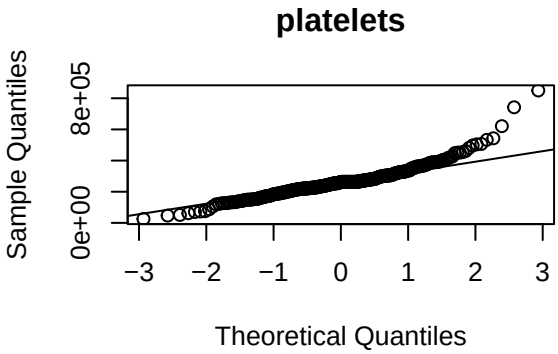



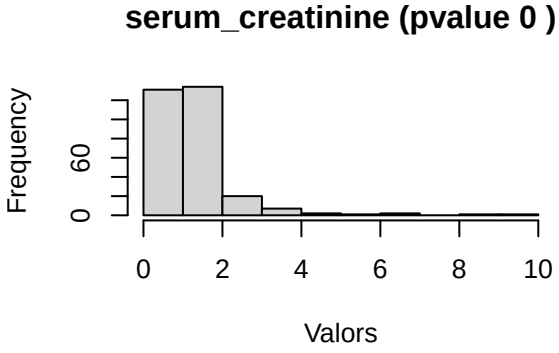



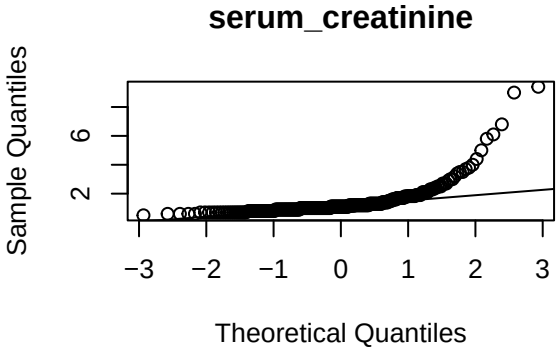

ejection_fraction (pvalue 0 )
Frequency
0
40
80
10
20
30
40
50
60
70
80
Valors

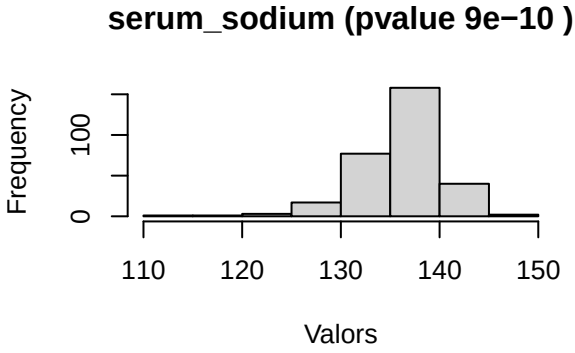

ejection_fraction
Sample Quantiles
20
50
80
−3
−2
−1
0
1
2
3
Theoretical Quantiles

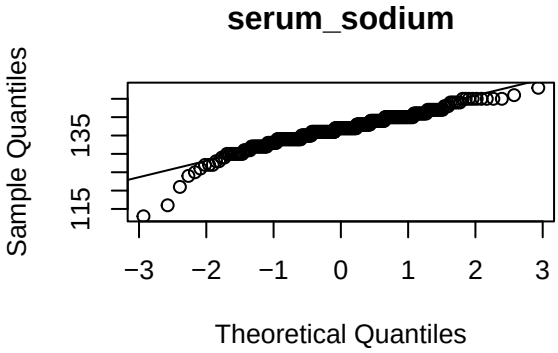

platelets (pvalue 2e−10 )
Frequency
0
100
0e+00
4e+05
8e+05
Valors

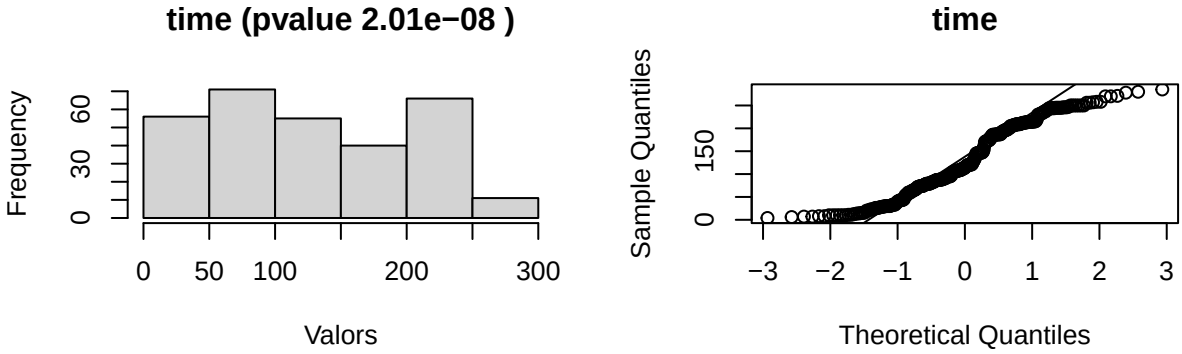

platelets
Sample Quantiles
0e+00
8e+05
−3
−2
−1
0
1
2
3
Theoretical Quantiles


serum_creatinine (pvalue 0 )
Frequency
0
60
0
2
4
6
8
10
Valors


serum_creatinine
Sample Quantiles
2
6
−3
−2
−1
0
1
2
3
Theoretical Quantiles


serum_sodium (pvalue 9e−10 )
Frequency
0
100
110
120
130
140
150
Valors


serum_sodium
Sample Quantiles
115
135
−3
−2
−1
0
1
2
3
Theoretical Quantiles




Observem que tot i que tenim variables que no es distribueixen segons una distribució normal, podem assumir que la mitjana mostral d'aquestes segueix una distribució normal ja que tenim una mostra de grandària superior a 30 registres (299 concretament) i pel teorema del límit central ho podem assumir.

## 4.3 Aplicació de proves estadístiques per comparar els grups de dades.

**En funció deles dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesis, correlacions, regressions, etc. Aplicar almenys tres mètodes d'anàlisi diferents.**

### 4.3.1 Contrast d'hipòtesis

El primer que volem comprovar és si hi ha diferències d'edat entre els homes i les dones quan moren. Per fer-ho farem una prova de contrast d'hipòtesis, en que la hipòtesis nul · la serà que la mitjana d'edat en que moren les dones és igual a la mitjana d'edat en que moren els homes. La hipòtesis alternativa és que la mitjana d'edat en que moren les dones no és igual a la mitjana d'edat en que moren els homes.

$H_0 : \mu_1 = \mu_2$

$H_1 : \mu_1 \neq \mu_2$

Per saber quin test hem d'aplicar, comparem les variàncies de les dues mostres.

```
# Fem la separació entre homes i dones
d_d <- dset$age[dset$DEATH_EVENT == TRUE & dset$sex == 'woman']
d_h <- dset$age[dset$DEATH_EVENT == TRUE & dset$sex == 'man']

# Comp
var.test(d_d, d_h)
```

```
##
##  F test to compare two variances
##
## data:  d_d and d_h
## F = 0.83537, num df = 33, denom df = 61, p-value = 0.5827
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4677973 1.5767422
## sample estimates:
## ratio of variances
##          0.8353672
```

El contrast de variàncies ens mostra un pvalor major de 0.05, per tant podem assumir la igualtat de variàncies en les dues poblacions.

Per tant aplicarem el test de la mitjana de dues poblacions independents, ja que les variables no estan relacionades, amb variància desconeguda igual i bilateral.

```r
# Apliquem el test
t.test(d_d, d_h, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  d_d and d_h
## t = -1.6803, df = 94, p-value = 0.09623
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10.2418387   0.8528444
## sample estimates:
## mean of x mean of y
##  62.17647  66.87097
```

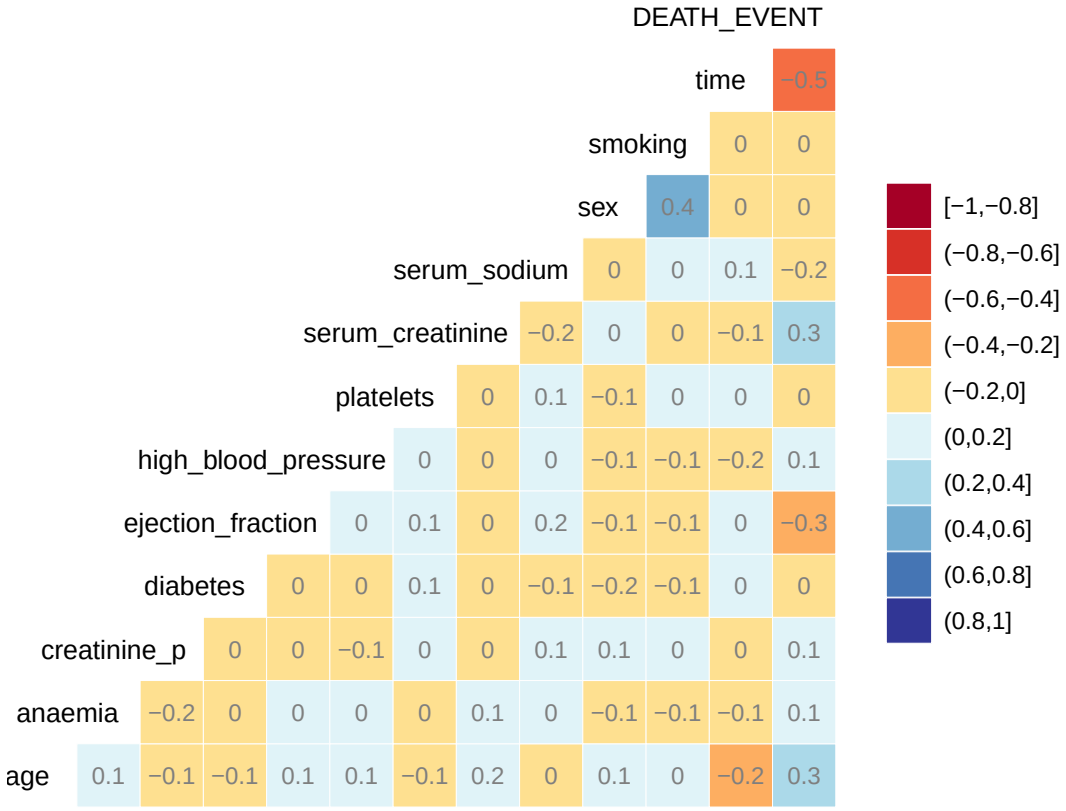
El pvalor (0.09623) és superior al nivell de significació (0.05), per tant podem acceptar la hipòtesis nul · la i concloure que en mitjana els homes i les dones moren a la mateixa edat amb un nivell de confiança del 95%.

## 4.4 Regressió lineal múltiple

Creem un model de regressió lineal múltiple per tal de predir si un pacient sobreviurà o morirà en funció d'unes determinades variables.

Abans creem una matriu de correlació per saber quines són les variables que tenen una relació més forta amb la variable DEATH_EVEN.

```r
# Mirem la matriu de correlacions
ggcorr(dset_2, nbreaks = 10, palette = 'RdYlBu', geom = 'tile', type='lower',
       label = TRUE, label_size = 3, label_color =  "grey50", hjust = 0.9, size = 3.5)
```



Analitzant les correlacions entre les diferents variables, observem que la més forta és en el temps de seguiment i si el pacient mor o no. La següent és entre el sexe i si fuma o no. Si que podem observar que la variable que té un grau de correlació més fort és la DEATH_EVEN, tot i que segueix siguent molt baix.

Separem el joc de dades en dos grups, un d'entrenament i l'altre de test. El 80% del registres seran pel d'entrenament i el 20% restant pel tes.

```
# Mantenim sempre els mateixos valors aleatòris
set.seed(121)

# Dividim el dataframe enamb la ratio que volem
sample = sample.split(dset,SplitRatio = 0.8)

# Creem el susbsets
train_ds =subset(dset,sample ==TRUE)
test_ds=subset(dset, sample==FALSE)
```

Creem el model amb les variables que tenen una correlació més forta amb la variable DEATH_EVENT.

```
#Creem el model
mod <- lm(DEATH_EVENT~age + ejection_fraction + serum_creatinine + time +
            serum_sodium, data = train_ds)
summary(mod)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time + serum_sodium, data = train_ds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77220 -0.27232 -0.03619  0.23926  1.00119
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.0569644  0.7207622   2.854 0.004713 **
## age                0.0046698  0.0020210   2.311 0.021738 *
## ejection_fraction -0.0085066  0.0020250  -4.201 3.81e-05 ***
## serum_creatinine   0.0858437  0.0220132   3.900 0.000126 ***
## time              -0.0027650  0.0003124  -8.852 2.35e-16 ***
## serum_sodium      -0.0107630  0.0052470  -2.051 0.041376 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3619 on 230 degrees of freedom
## Multiple R-squared:  0.4024, Adjusted R-squared:  0.3894
## F-statistic: 30.97 on 5 and 230 DF,  p-value: < 2.2e-16
```

Observem per una banda que el model és vàlid ja amb un nivell de confiança del 95%, ja que el seu pvalue és inferior al nivell de significància. Per altre banda totes les variables aporten al model ja que tenen un pvalue inferior al nivell de significància. Per últim el coeficient de determinació és de $R^2 = 0.4024$, és a dir que el model de regressió lineal múltiple ens explica el 40,24% de la variància de les observacions.

## 4.5 Arbres de decisió

El primer que farem serà adequar el dataframe, seleccionant les columnes en que els valors son factors o logic. En el cas de les columnes que contenen dades logic, les convertirem a factors.

```r
dset_tree <- dset[c("segment_age", "segment_ejection", "segment_creatine_p",
                    "segment_platelets", "segment_serum_creatinine",
                    "segment_serum_sodium", "diabetes", "high_blood_pressure",
                    "sex", "smoking", "anaemia", "DEATH_EVENT")]


# Converteixo a factorial les columnes logic
dset_tree$anaemia <- as.factor(dset_tree$anaemia)
levels(dset_tree$anaemia)[match("FALSE",levels(dset_tree$anaemia))] <- "no_anaemia"
levels(dset_tree$anaemia)[match("TRUE",levels(dset_tree$anaemia))] <- "si_anaemia"

dset_tree$diabetes <- as.factor(dset_tree$diabetes)
levels(dset_tree$diabetes)[match("FALSE",levels(dset_tree$diabetes))] <- "no_diabetes"
levels(dset_tree$diabetes)[match("TRUE",levels(dset_tree$diabetes))] <- "si_diabetes"

dset_tree$high_blood_pressure <- as.factor(dset_tree$high_blood_pressure)
levels(dset_tree$high_blood_pressure)[match(
  "FALSE",levels(dset_tree$high_blood_pressure))] <- "no_high_blood_pressure"
levels(dset_tree$high_blood_pressure)[match(
  "TRUE",levels(dset_tree$high_blood_pressure))] <- "si_high_blood_pressure"

dset_tree$smoking <- as.factor(dset_tree$smoking)
levels(dset_tree$smoking)[match("FALSE",levels(dset_tree$smoking))] <- "no_smoking"
levels(dset_tree$smoking)[match("TRUE",levels(dset_tree$smoking))] <- "si_smoking"

dset_tree$DEATH_EVENT <- as.factor(dset_tree$DEATH_EVENT)
levels(dset_tree$DEATH_EVENT)[match("FALSE",levels(dset_tree$DEATH_EVENT))] <- "NO_DEATH_EVENT"
levels(dset_tree$DEATH_EVENT)[match("TRUE",levels(dset_tree$DEATH_EVENT))] <- "SI_DEATH_EVENT"

# Observem el nou dataset
summary(dset_tree)
```

```
##  segment_age      segment_ejection     segment_creatine_p     segment_platelets
##  40-49:47    ej_molt-baix : 5      cp_baix      : 75      pl_molt-baix : 8
##  50-59:82    ej_baix      :54      cp_mitja-baix:113      pl_baix      :67
##  60-69:93    ej_mitja-baix:83      cp_mitja-alt : 75      pl_mitja-baix:76
##  70-79:52    ej_mitja-alt :77      cp_alt       : 18      pl_mitja-alt :73
##  80-95:25    ej_alt       :44      cp_motl-alt  : 18      pl_alt       :72
##              ej_motl-alt  :36                             pl_motl-alt  : 3
##   segment_serum_creatinine    segment_serum_sodium        diabetes
##  sc_baix      : 49         ss_motl-baix :  2       no_diabetes:174
##  sc_mitja-baix: 82         ss_baix      : 25       si_diabetes:125
##  sc_mitja-alt :101         ss_mitja-baix: 72
##  sc_alt       : 59         ss_mitja-alt :123
##  sc_motl-alt  :  8         ss_alt       : 77
##
##              high_blood_pressure    sex            smoking          anaemia
##  no_high_blood_pressure:194      woman:105   no_smoking:203   no_anaemia:170
##  si_high_blood_pressure:105      man  :194   si_smoking: 96   si_anaemia:129
##
##
##
##
##          DEATH_EVENT
```

```
##  NO_DEATH_EVENT:203
##  SI_DEATH_EVENT: 96
##
##
##
##
```

A l'hora de preprar les dades fer crear l'arbre de decissió, el primer que hem de fer és dividir el joc de dades en dos parts. Una part d'entrenament i l'altre de prova. És a dir que utilitzarem una part del joc de dades per construir l'abre de decisió i l'altre per evaluar-lo. El grup d'entrenament tindrà 2/3 del joc de dades i el grup de prova 1/3. La variable que ens classificarà el joc de dades serà l'anomenada 'DEATH_EVENT'. En funció d'això, el primer que fem és crear dos varaibles noves, una amb els valors de la columna de la variable 'default' i l'altre amb els de la resta de columnes.

```
y <- dset_tree[,12]
X <- dset_tree[,1:11]
```

Per dividir el joc de dades en els dos grups, podem definir una manera de separar les dades en funció d'un paràmetre, en aquest cas del "split_prop". Com que volem que el grup d'entrenament tingui 2/3 de les files i el grup test 1/3 de les dades, dividirem el conjunt en tres parts.

```
set.seed(1236)

# Li creem la variable amb el número que volem dividir el grup
split_prop <- 3

# Calculem els index que ens serveixen per seleccionar les files que van en cada grup
indexes = sample(1:nrow(dset_tree), size=floor(((split_prop-1)/split_prop)*nrow(dset_tree)))

# Creem les variables amb els grups d'entrenament i de test
trainX<-X[indexes,]
trainy<-y[indexes]
testX<-X[-indexes,]
testy<-y[-indexes]
```

Un cop creats els grups creem l'arbre de decisió. Abans confirmem que la variable trainy sigui de tipus factor.

```
# Assegurem que la variable trainy sigui de tipus factor
trainy = as.factor(trainy)

# Creem el model
model <- C50::C5.0(trainX, trainy,rules=TRUE )
summary(model)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Jan  4 20:03:46 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 199 cases (12 attributes) from undefined.data
##
## Rules:
```

```
##
## Rule 1: (123/18, lift 1.3)
##  segment_age in {40-49, 50-59, 60-69, 70-79}
##  segment_ejection in {ej_mitja-baix, ej_mitja-alt, ej_alt, ej_motl-alt}
##  segment_serum_creatinine in {sc_baix, sc_mitja-baix, sc_mitja-alt}
##  ->  class NO_DEATH_EVENT  [0.848]
##
## Rule 2: (92/23, lift 1.1)
##  high_blood_pressure = no_high_blood_pressure
##  sex = man
##  ->  class NO_DEATH_EVENT  [0.745]
##
## Rule 3: (21/6, lift 2.1)
##  segment_ejection in {ej_molt-baix, ej_baix}
##  segment_serum_creatinine in {sc_baix, sc_mitja-baix, sc_mitja-alt}
##  segment_serum_sodium in {ss_baix, ss_mitja-alt, ss_alt}
##  ->  class SI_DEATH_EVENT  [0.696]
##
## Rule 4: (16/5, lift 2.0)
##  segment_age = 80-95
##  segment_ejection in {ej_mitja-baix, ej_mitja-alt, ej_alt, ej_motl-alt}
##  ->  class SI_DEATH_EVENT  [0.667]
##
## Rule 5: (40/13, lift 2.0)
##  segment_serum_creatinine in {sc_alt, sc_motl-alt}
##  ->  class SI_DEATH_EVENT  [0.667]
##
## Rule 6: (42/14, lift 2.0)
##  segment_ejection in {ej_molt-baix, ej_baix}
##  ->  class SI_DEATH_EVENT  [0.659]
##
## Default class: NO_DEATH_EVENT
##
##
## Evaluation on training data (199 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       6   40(20.1%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     117    15    (a): class NO_DEATH_EVENT
##      25    42    (b): class SI_DEATH_EVENT
##
##
##  Attribute usage:
##
##   92.46% segment_serum_creatinine
##   90.95% segment_ejection
##   69.85% segment_age
```

```
##   46.23% high_blood_pressure
##   46.23% sex
##   10.55% segment_serum_sodium
##
##
## Time: 0.0 secs
```

```
# Grau d'influència de les variables.
importancia_usage <- C50::C5imp(model, metric = "usage")
importancia_splits <- C50::C5imp(model, metric = "splits")
importancia_usage
```

```
##                          Overall
## segment_serum_creatinine   92.46
## segment_ejection           90.95
## segment_age                69.85
## high_blood_pressure        46.23
## sex                        46.23
## segment_serum_sodium       10.55
## segment_creatine_p          0.00
## segment_platelets           0.00
## diabetes                    0.00
## smoking                     0.00
## anaemia                     0.00
```

```
importancia_splits
```

```
##                           Overall
## segment_ejection         33.333333
## segment_serum_creatinine 25.000000
## segment_age              16.666667
## high_blood_pressure       8.333333
## segment_serum_sodium      8.333333
## sex                       8.333333
## segment_creatine_p        0.000000
## segment_platelets         0.000000
## diabetes                  0.000000
## smoking                   0.000000
## anaemia                   0.000000
```

El primer que podem observar és que hi ha un 20,1% de les files que és classiíquen de forma errornia. En termes absoluts representa que l'arbre classifica malament 40 de les 199 files.

Ens ha creat 6 regles:

segment_age in {40-49, 50-59, 60-69, 70-79} + segment_ejection in {ej_mitja-baix, ej_mitja-alt, ej_alt} + segment_serum_creatinine in {sc_baix, sc_mitja-baix, sc_mitja-alt} -> NO_DEATH_EVENT. Validesa 84,6%

high_blood_pressure = no_high_blood_pressure + sex = man -> NO_DEATH_EVENT. Validesa 74,5%

segment_ejection in {ej_molt-baix, ej_baix} + segment_serum_creatinine in {sc_baix, sc_mitja-baix, sc_mitja-alt} + segment_serum_sodium in {ss_baix, ss_mitja-alt, ss_alt} -> SI_DEATH_EVENT. Validesa 69,6%

segment_age = 80-95 + segment_ejection in {ej_mitja-baix, ej_mitja-alt, ej_alt, ej_motl-alt} -> SI_DEATH_EVENT. Validesa 66,7%

segment_serum_creatinine in {sc_alt, sc_motl-alt} -> SI_DEATH_EVENT. Validesa 66,7%

segment_ejection in {ej_molt-baix, ej_baix} -> SI_DEATH_EVENT. Validesa 65,9%

Podem observar que per realitzar les regles utilitza a sis variables. Sent el “segment_serum_creatinine” amb un 92,46% i el ‘segment_ejection’ en un 90,95% les més utilitzades.

# 5 Representació dels resultats a partir de taules i gràfiques.

Apliquem al grup test el model lineal per poder avaluar-ne el seu funcionament.

```
# Fem la predicció
pred1 <- predict(mod, test_ds)

# Li diem que ens classifiqui la predicció entre 1 i 0. En que 1 serà si el
#valor és igual o superior a 0.5
pred1_clas <- ifelse(pred1>=0.5, "TRUE","FALSE")

# Creem un dataframe amb les dades observades i les prediccions que ha fet
performance_data<-data.frame(observat=test_ds$DEATH_EVENT,
           predit= pred1_clas)
```

Calculem els valors de la matriu

```
# Numero de registres
total <- nrow(performance_data)

# Calculem els valors de dins de la matriu
# vp(verdader postiu), vn(verdader negatiu), fp(fals positiu), fn(fals negatiu)
vp<-sum(performance_data$observat=="TRUE" & performance_data$predit=="TRUE")
vn<-sum(performance_data$observat=="FALSE" & performance_data$predit=="FALSE")
fp<-sum(performance_data$observat=="FALSE" & performance_data$predit=="TRUE")
fn<-sum(performance_data$observat=="TRUE" & performance_data$predit=="FALSE")
# imprimim els valors
data.frame(vp,vn,fp,fn)
```

```
##   vp vn fp fn
## 1 16 37  3  7
```

```
# Calculem els valors totals
positiu <- sum(performance_data$observat=="TRUE")
negatiu <- sum(performance_data$observat=="FALSE")
predit_positiu <- sum(performance_data$predit=="TRUE")
predit_negatiu <- sum(performance_data$predit=="FALSE")
# Ho mostrem
data.frame(positiu, negatiu,predit_positiu,predit_negatiu)
```

```
##   positiu negatiu predit_positiu predit_negatiu
## 1      23      40             19             44
```

```
# Creem la matriu
m_conf <- cbind(c(vn, fn, predit_negatiu), c(fp, vp, predit_positiu),
                c(negatiu, positiu, total))
colnames(m_conf) <- c('prob predit < 50%', 'prob predit >= 50%', 'Total')
rownames(m_conf) <- c('Observat: FALSE', 'observat: TRUE', 'Total')
m_conf
```

```
##                  prob predit < 50% prob predit >= 50% Total
## Observat: FALSE                 37                  3    40
## observat: TRUE                   7                 16    23
## Total                           44                 19    63
```

```
# Calculem els valors que deriben de la taula
exactitut <- (vp+vn)/total
taxa_error <- (fp+fn)/total
```

```
sensibilitat <- vp/positiu
especificitat <- vn/negatiu
precisio <- vp/predit_positiu
npv <- vn / predit_negatiu
data.frame(exactitut,taxa_error,sensibilitat,especificitat,precisio,npv)
```

```
##   exactitut taxa_error sensibilitat especificitat  precisio       npv
## 1 0.8412698  0.1587302    0.6956522         0.925 0.8421053 0.8409091
```

El primer que podem observar és que el percentatge de casos encertats és del 84%, que és un nombre elevat. Si analitzem la sensibilitat, que ens diu quin percentatge representen els positius (morts) predits entre tots els positius, observem que el 69% dels positius han estat predits. Per altra banda si analitzem la especificitat, que ens diu quin percentatge representen els negatius (no morts) predits entre tots els negatius, observem que el 92% dels negatius han estat predit. El que ens ve a dir que el model té un percentatge d'encert elevat per predir si el pacient no morirà, però si ha de predir si el pacient morirà el percentatge ja és més baix.

Per altra banda el model d'arbre de decisió al grup de dades test per analitzar la qualitat del model predint si els pacients es moren o sobreviuen.

```
# Fem la predicció
predicted_model <- predict(model, testX, type="class" )
print(sprintf("La precisió de l'arbre és: %.4f %%",100*sum(predicted_model == testy)
              / length(predicted_model)))
```

```
## [1] "La precisió de l'arbre és: 79.0000 %"
```

Observem que la precisió del model és del 79%. És a dir que quan li entroduïm les dades d'un pacient, té un 79% de possibilitats d'encertar si sobreviurà o no.

Creant una matriu de confusió, podrem veure a on s'ubiquen els errors i encerts.

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

```
##                 Predicted
## testy            NO_DEATH_EVENT SI_DEATH_EVENT
##   NO_DEATH_EVENT             60             11
##   SI_DEATH_EVENT             10             19
```

Podem visualitzar aquesta matriu en valors percentuals, que ens serà més fàcil d'interpretar.

```
CrossTable(testy, predicted_model,prop.chisq  = FALSE, prop.c = FALSE,
           prop.r =FALSE,dnn = c('Reality', 'Prediction'))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                | Prediction
##        Reality | NO_DEATH_EVENT | SI_DEATH_EVENT |      Row Total |
```

```
## ---------------|----------------|----------------|----------------|
## NO_DEATH_EVENT |             60 |             11 |             71 |
##                |          0.600 |          0.110 |                |
## ---------------|----------------|----------------|----------------|
## SI_DEATH_EVENT |             10 |             19 |             29 |
##                |          0.100 |          0.190 |                |
## ---------------|----------------|----------------|----------------|
##   Column Total |             70 |             30 |            100 |
## ---------------|----------------|----------------|----------------|
##
##
```

Observem que el model tendeix a infrarrepresentar a les persones que sobreviuen i per tant a sobrerrepresenatar a les que moren. Tot i que en percentatges molt petits. Observem que a la predicció hi ha un 70% de persones que sobreviuen i en canvi a la realitat n'hi ha un 71%. En canvi prediu que hi ha un 30% de persones que moren i al grup test n'hi ha un 29%.

# 6 Resolució del problema.

**A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema?**

Per una banda podem concloure que en mitjana els homes i les dones moren a la mateixa edat amb un nivell de confiança del 95%.

Per una altre banda a partir del model lineal múltiple tenim un percentatge d'encert elevat per predir si el pacient no morirà(92%), però si ha de predir si el pacient morirà el percentatge ja és més baix (69%), la qual cosa només en serviria per detectar si el pacient sobreviurà.

Per acabar, el model d'arbre de decisió té una precissió acceptable 80% i tendeix a infrarrepresentar a les persones que sobreviuen i per tant a sobrerrepresenatar a les que moren. Tot i que en percentatges molt petits.

# 7 Codi.

**Cal adjuntar el codi, preferiblement en R, amb el que s'ha realitzat la neteja, anàlisi i representació de les dades. Si ho preferiu, també podeu treballar en Python.**

No adjuntem el codi, ja que el codi ja està present en el propi document.

# 8 Taula de contribucions

```
tab <- matrix(c('Investigació previa', 'Redacció de les respostes',
                'Desenvolupament codi', 'MPS','MPS','MPS'), ncol = 2)
colnames(tab) <- c('Contribucions', 'Firma')
kable(as.table(tab), row.names = F)
```

| Contribucions | Firma |
|---|---|
| Investigació previa | MPS |
| Redacció de les respostes | MPS |
| Desenvolupament codi | MPS |