Pràctica 2

Pol Moya Betriu i Xavier Martin Bravo

20 de December, 2020

Índex

1 Descripció del dataset

Després de valorar diversos jocs de dades que semblaven interessants i valorar que aquests eren aptes per a la realització de la pràctica, ens hem decantat per un sobre persones que han sofert infarts. Aquest joc de dades té vàries característiques que fan que pugui ser un bon model per a aplicar algoritmes supervisats, algoritmes no supervisats i regles d'associació.

Les malalties cardiovasculars són la principal causa de mort globalment (fins al 30%), moren aproximadament 17.9 milions de persones cada any. L'atac de cor és una de les principals conseqüències, causades per les malalties cardiovasculars, aquest joc de dades conté 12 atributs que es poden utilitzar per a predir la mortalitat dels atacs de cor. (Organización Mundial de la Salud 2018)

La majoria dels atacs de cor es poden prevenir millorant certs hàbits, com per exemple fumar, una mala dieta, l'obesitat, el sedentarisme i l'alcoholisme.

L'objectiu del joc de dades és intentar predir si persones de risc, han mort o no d'un atac de cor en un període determinat de temps en els quals estan en seguiment. D'aquesta manera una detecció precoç i una bona gestió d'un posible atac de cor, poden salvar moltes de les vides d'aquestes persones amb un risc més alt.

Descripció de les variables contingudes al joc de dades:

- **age:** integer que descriu edat del pacient (anys).
- **anaemia:** factor que especifica si el pacient pateix anèmia o no.
- **creatinine_phosphokinase:** integer que representa el nivell de *CPK* en la sang en (*mcg/L*).
- **diabetes:** factor que indica si el pacient és diabètic o no.
- **ejection_fraction :** integer que descriu el percentatge de sang que surt del cor en cada contracció en (%).
- **high_blood_pressure:** factor que indica si el pacient té hipertensió o no.
- **platelets:** numeric que representa el nombre de plaquetes en sang del pacient (kiloplatelets/mL).
- **serum_creatinine:** numeric que representa el nivell de creatinina en la sang (mg/dL).
- **serum_sodium:** integer que indica el nivell de sodi en sang (mEq/L).
- **sex:** factor que indica si el sexe del pacient és masculí o femení.
- **smoking:** factor que indica si el pacient fuma o no.
- **time:** integer període de seguiment en dies.
- **DEATH_EVENT:** factor que indica si el pacient ha mort o no durant el període de seguiment.

2 Integració i selecció de les dades d'interès a analitzar.

Un cop hem vist els atributs del joc dades, determinem que tots són d'interès perquè no sabem quins atributs influeixen a l'hora de predir si una persona es morirà o no d'un atac de cor. Per tant de moment utilitzarem tots els atributs inclosos en el joc de dades i més endavant ja conclourem si tots són significatius o realment hi ha algun atribut que és prescindible per aquesta anàlisi.

3 Neteja de les dades.

Carreguem el joc de dades i comprovem que aquest s'ha llegit de forma correcta:

```
heart_data<-read.csv("./heart_failure_clinical_records_dataset.csv", header=T, sep=",")
#Comprovem que s'ha llegit correctament
head(heart_data)
```

```
##   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
```

```
#Comprovem que la dimensió és la correcta
dim(heart_data)
```

```
## [1] 299  13
```

3.1 Anàlisi exploratòria del joc de dades

```
#Visualitzem els tipus de les variables
str(heart_data)
```

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

Observem que tenim variables que no estan en el format que haurien d'estar. Per tant, abans de continuar amb l'anàlisi exploratòria, transformarem les binaries i booleanes a categòriques per tal de visualitzar millor com estan distribuïdes les dades. Ens guardarem una còpia del joc de dades per a poder recuperar la transfromació, ja que la majoria d'eines i models per analitzar dades només accepten números com a valors de les variables.

```
#Realizem una còpia
heart_data1 <- heart_data
#anaemia
i <- heart_data$anaemia == '1'
heart_data$anaemia[i] <- "Si"
i <- heart_data$anaemia == '0'
heart_data$anaemia[i] <- "No"
heart_data$anaemia <- as.factor(heart_data$anaemia)
#diabetes
```

```r
i <- heart_data$diabetes == '1'
heart_data$diabetes[i] <- "Si"
i <- heart_data$diabetes == '0'
heart_data$diabetes[i] <- "No"
heart_data$diabetes <- as.factor(heart_data$diabetes)
#high_blood_pressure
i <- heart_data$high_blood_pressure == '1'
heart_data$high_blood_pressure[i] <- "Si"
i <- heart_data$high_blood_pressure == '0'
heart_data$high_blood_pressure[i] <- "No"
heart_data$high_blood_pressure <- as.factor(heart_data$high_blood_pressure)
#sex
i <- heart_data$sex == '0'
heart_data$sex[i] <- "femeni"
i <- heart_data$sex == '1'
heart_data$sex[i] <- "masculi"
heart_data$sex <- as.factor(heart_data$sex)
#smoking
i <- heart_data$smoking == '1'
heart_data$smoking[i] <- "Si"
i <- heart_data$smoking == '0'
heart_data$smoking[i] <- "No"
heart_data$smoking <- as.factor(heart_data$smoking)
#DEATH_EVENT
i <- heart_data$DEATH_EVENT == '1'
heart_data$DEATH_EVENT[i] <- "Si"
i <- heart_data$DEATH_EVENT == '0'
heart_data$DEATH_EVENT[i] <- "No"
heart_data$DEATH_EVENT <- as.factor(heart_data$DEATH_EVENT)
#Observem les transformacions
str(heart_data)
```

```
## 'data.frame':    299 obs. of  13 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : Factor w/ 2 levels "No","Si": 1 1 1 2 2 2 2 2 1 2 ...
##  $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes                : Factor w/ 2 levels "No","Si": 1 1 1 1 2 1 1 2 1 1 ...
##  $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : Factor w/ 2 levels "No","Si": 2 1 1 1 1 2 1 1 1 2 ...
##  $ 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 "femeni","masculi": 2 2 2 2 1 2 2 2 1 2 ...
##  $ smoking                 : Factor w/ 2 levels "No","Si": 1 1 2 1 1 2 1 2 1 2 ...
##  $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
##  $ DEATH_EVENT             : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
```

A continuació visualitzarem una descriptiva de les variables amb la funció *summary* (resum) de les dades on s'aprecia la transformació de les variables categòriques per tal de poder realitzar una millor anàlisi exploratòria:

```r
#Resum de les dades
summary(heart_data)
```

```
##       age        anaemia  creatinine_phosphokinase diabetes ejection_fraction
```

```
##  Min.   :40.00   No:170   Min.   :  23.0          No:174   Min.   :14.00
##  1st Qu.:51.00   Si:129   1st Qu.: 116.5          Si:125   1st Qu.:30.00
##  Median :60.00            Median : 250.0                   Median :38.00
##  Mean   :60.83            Mean   : 581.8                   Mean   :38.08
##  3rd Qu.:70.00            3rd Qu.: 582.0                   3rd Qu.:45.00
##  Max.   :95.00            Max.   :7861.0                   Max.   :80.00
##  high_blood_pressure   platelets      serum_creatinine  serum_sodium
##  No:194              Min.   : 25100   Min.   :0.500    Min.   :113.0
##  Si:105              1st Qu.:212500   1st Qu.:0.900    1st Qu.:134.0
##                      Median :262000   Median :1.100    Median :137.0
##                      Mean   :263358   Mean   :1.394    Mean   :136.6
##                      3rd Qu.:303500   3rd Qu.:1.400    3rd Qu.:140.0
##                      Max.   :850000   Max.   :9.400    Max.   :148.0
##       sex      smoking       time       DEATH_EVENT
##  femeni :105   No:203   Min.   :  4.0   No:203
##  masculi:194   Si: 96   1st Qu.: 73.0   Si: 96
##                         Median :115.0
##                         Mean   :130.3
##                         3rd Qu.:203.0
##                         Max.   :285.0
```

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

A continuació estudiarem i tractarem els valors buits (NA):

```
#Estudiem els valors buits
colSums(is.na(heart_data))
```

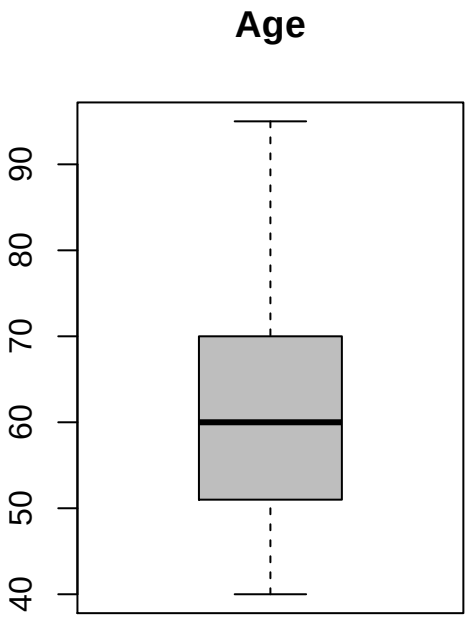
```
##                      age                  anaemia creatinine_phosphokinase
##                        0                        0                        0
##                 diabetes        ejection_fraction      high_blood_pressure
##                        0                        0          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

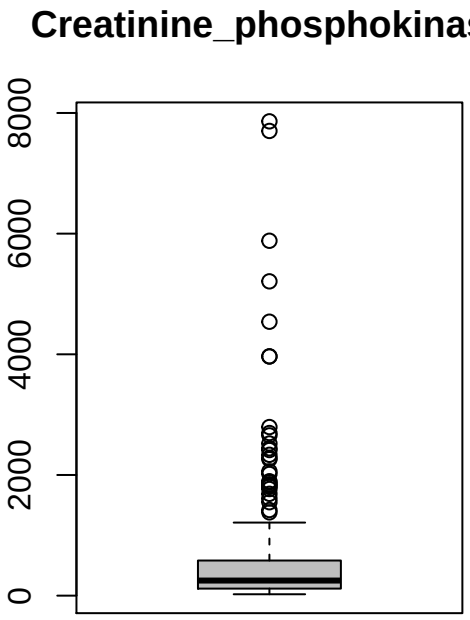
Com podem observar, no hi ha valors buits al joc de dades per tant podem seguir endavant.

3.3 Identificació i tractament de valors extrems.

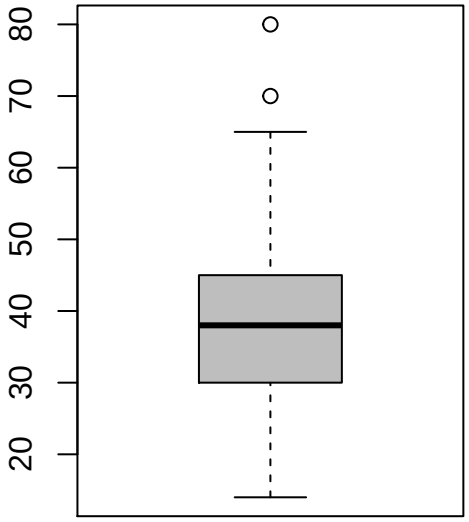
Visualitzarem els valors extrems o *outliers* amb un boxplot per cada variable numèrica.

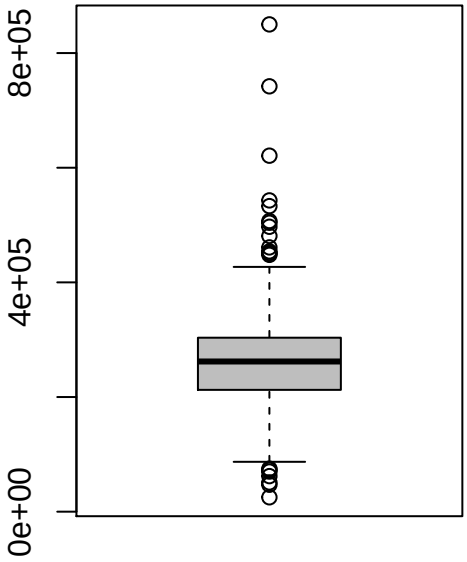
```
#Estudiem els outliers
attach(heart_data)
par(mfrow=c(1,2))
boxplot(age,main="Age", col="gray")
boxplot(creatinine_phosphokinase,main="Creatinine_phosphokinase", col="gray")
```



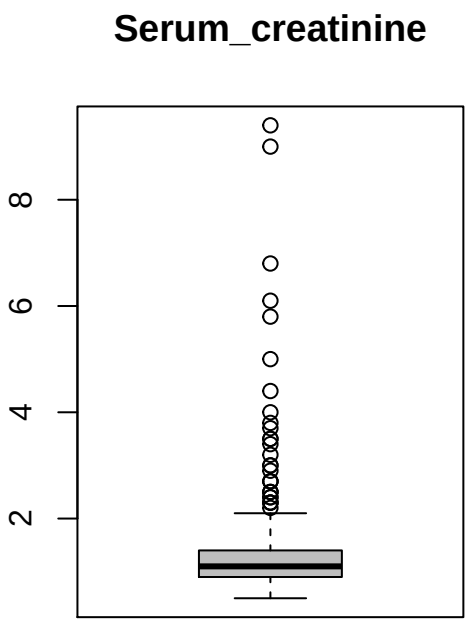


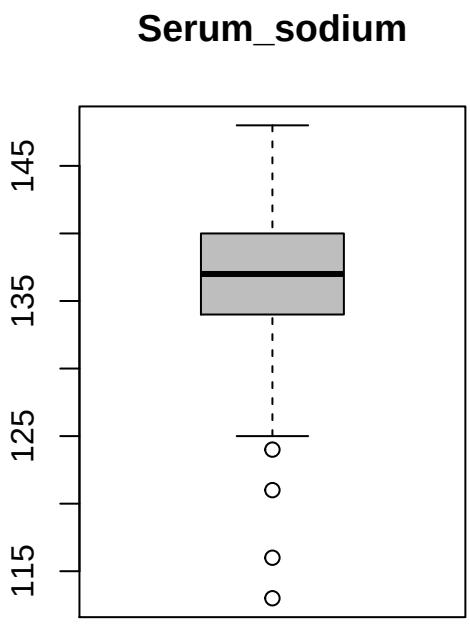

```
boxplot(ejection_fraction,main="Ejection_fraction", col="gray")
boxplot(platelets,main="Platelets", col="gray")
```



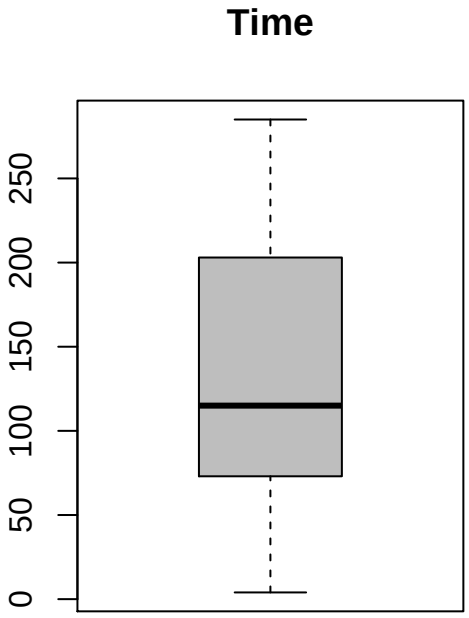



```
boxplot(serum_creatinine,main="Serum_creatinine", col="gray")
boxplot(serum_sodium,main="Serum_sodium", col="gray")
```






```
boxplot(time,main="Time", col="gray")
```



Podem veure que totes les variables menys age i time tenen outliers. A continuació observarem quins són aquests per a cada variable.

```
#Visualitzem els outliers
#creatinine_phosphokinase
summary(heart_data$creatinine_phosphokinase)
```

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

```
boxplot.stats(heart_data$creatinine_phosphokinase)$out
```

```
##  [1] 7861 2656 1380 3964 7702 5882 5209 1876 1808 4540 1548 1610 2261 1846 2334
## [16] 2442 3966 1419 1896 1767 2281 2794 2017 2522 2695 1688 1820 2060 2413
```

```r
#ejection_fraction
summary(heart_data$ejection_fraction)
```

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

```r
boxplot.stats(heart_data$ejection_fraction)$out
```

```
## [1] 80 70
```

```r
#platelets
summary(heart_data$platelets)
```

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

```r
boxplot.stats(heart_data$platelets)$out
```

```
##  [1] 454000  47000 451000 461000 497000 621000 850000 507000 448000  75000
## [11]  70000  73000 481000 504000  62000 533000  25100 451000  51000 543000
## [21] 742000
```

```r
#serum_creatinine
summary(heart_data$serum_creatinine)
```

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

```r
boxplot.stats(heart_data$serum_creatinine)$out
```

```
##  [1] 2.7 9.4 4.0 5.8 3.0 3.5 2.3 3.0 4.4 6.8 2.2 2.7 2.3 2.9 2.5 2.3 3.2 3.7 3.4
## [20] 6.1 2.5 2.4 2.5 3.5 9.0 5.0 2.4 2.7 3.8
```

```r
#serum_sodium
summary(heart_data$serum_sodium)
```

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

```r
boxplot.stats(heart_data$serum_sodium)$out
```

```
## [1] 116 121 124 113
```

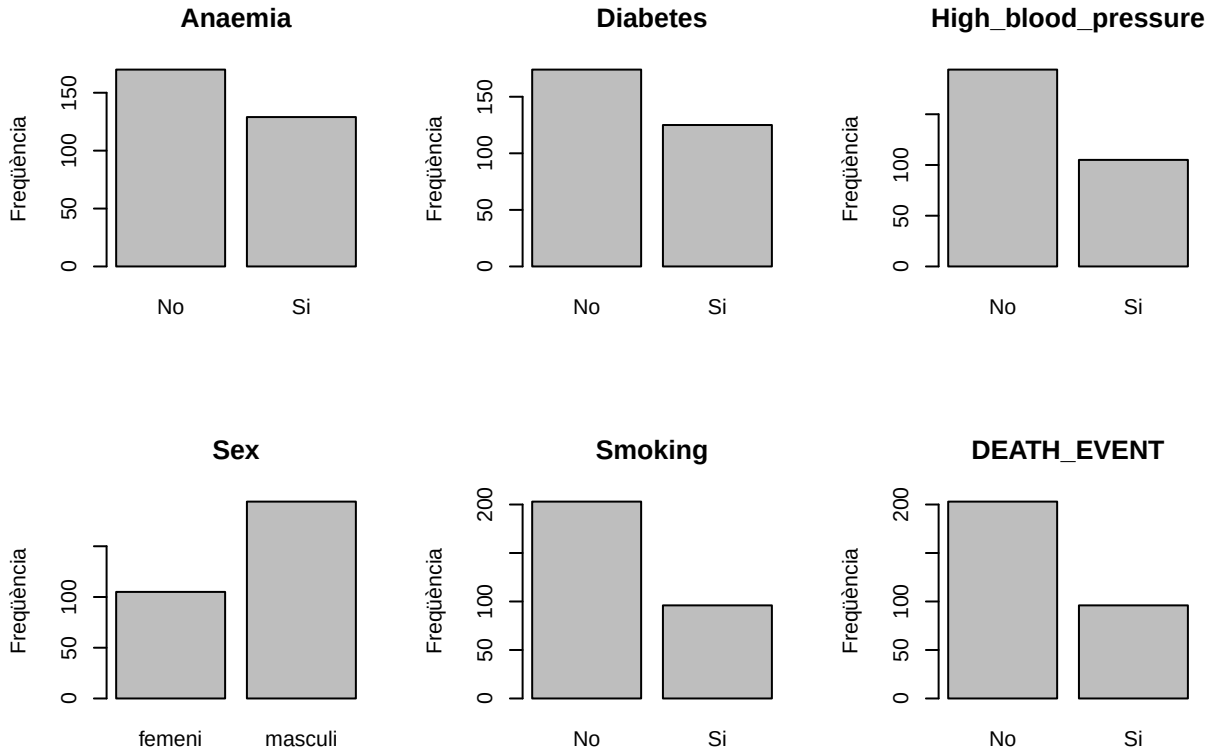
- *creatinine_phosphokinase*: observem que tenim bastanta quantitat de posibles outliers pel costat dret de la distribució i aquests separen molt de la mediana i la mitjana.
- *ejection_fraction*: només tenim 2 outliers i no són excessivament grans.
- *platelets*: en aquesta variable també tenim una gran quantitat d'outliers i n'hi ha pels dos costats de la distribució.
- *serum_creatinine*: tenim outliers que es separen molt de les variables centrals, per exemple la mediana és 1.1 i teni mun màxim de 9.4.
- *serum_sodium*: en aquesta variable només tenim 4 outliers i proporcionalment no s'allunyen tant com en la variable *serum_creatinine*.

Després de visualitzar els possibles *outliers* que presenta el nostre joc de dades, s'ha decidit no excloure'ls perquè molt probablement representen dades reals que després ens ajudaran a veure quan i perquè una persona és mor en un període de temps posterior a patir un infart.

3.3.1 Variables categòriques

Visualitzarem i analitzarem les distribucions de les variables categòriques:

```r
attach(heart_data)
par(mfrow=c(2,3))
barplot(table(anaemia), main = "Anaemia", ylab='Freqüència')
barplot(table(diabetes), main = "Diabetes", ylab='Freqüència')
barplot(table(high_blood_pressure), main = "High_blood_pressure", ylab='Freqüència')
barplot(table(sex), main = "Sex", ylab='Freqüència')
barplot(table(smoking), main = "Smoking", ylab='Freqüència')
barplot(table(DEATH_EVENT), main = "DEATH_EVENT", ylab='Freqüència')
```



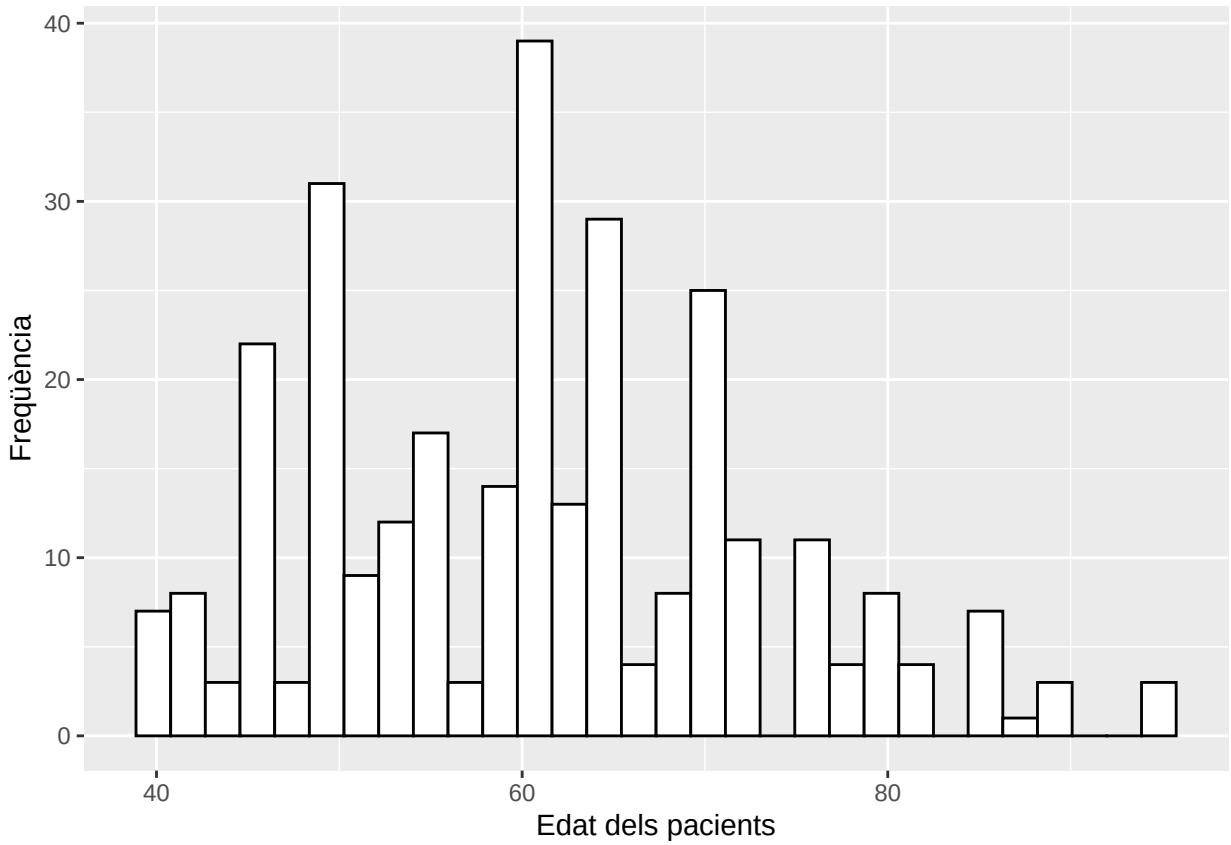
Podem observar que tenim:

- **anaemia:** observem que tenim més pacients no anèmics (170) que pacients anèmics (129).
- **diabetes:** hi ha menys pacients diabètics (125) que no diabètics (174).
- **high_blood_pressure:** hi ha menys pacients amb hipertensió que pacients que no en tenen, 105 i 194 respectivament.
- **sex:** en el joc de dades hi ha més persones amb sexe masculí (194) a femení (105).
- **smoking:**hi ha més persones no fumadores (203) que fumadores (96).
- **DEATH_EVENT:** hi ha més persones que sobreviuen que persones que moren en el període de seguiment 203 i 96 respectivament.

3.3.2 Variables numèriques

Un cop vistes les distribucions categòriques, passarem a analitzar les distribucions de les variables numèriques.

```r
library(ggplot2)
ggplot(heart_data, aes(x = age)) + geom_histogram(fill="white",colour="black") + ylab("Freqüència") + xl
```



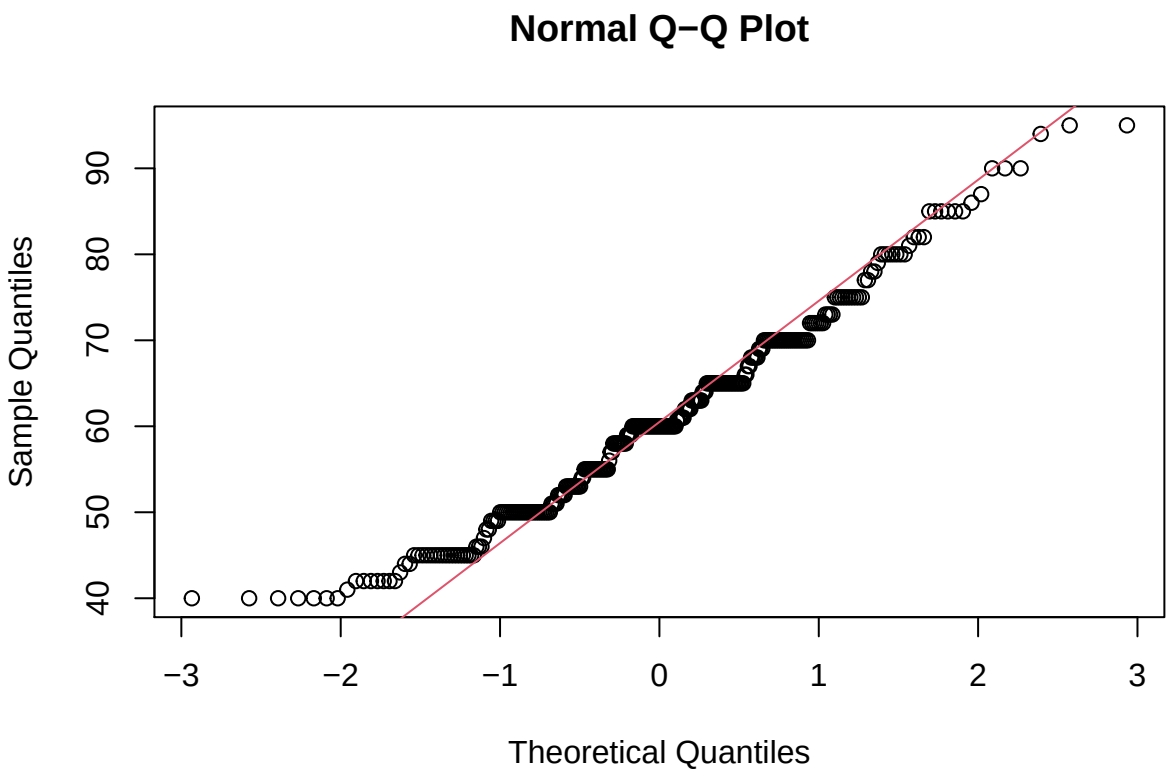
3.3.2.1 Age

Veiem que la distribució, no sembla que segueixi una distribució normal perquè tenim molts alts i baixos, tot i que sí que té forma de campana de Gauss. Definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat. Tant per aquesta variable, com les següents, assumirem un nivell de confiança del 95%. Per tant $\alpha = 0.05$.

H_0 : La variable *age* segueix una distribució normal.

H_1 : La variable *age* no segueix una distribució normal.

```
#Observem la distribució amb la funció qqnorm
qqnorm(heart_data$age);qqline(heart_data$age, col = 2)
```



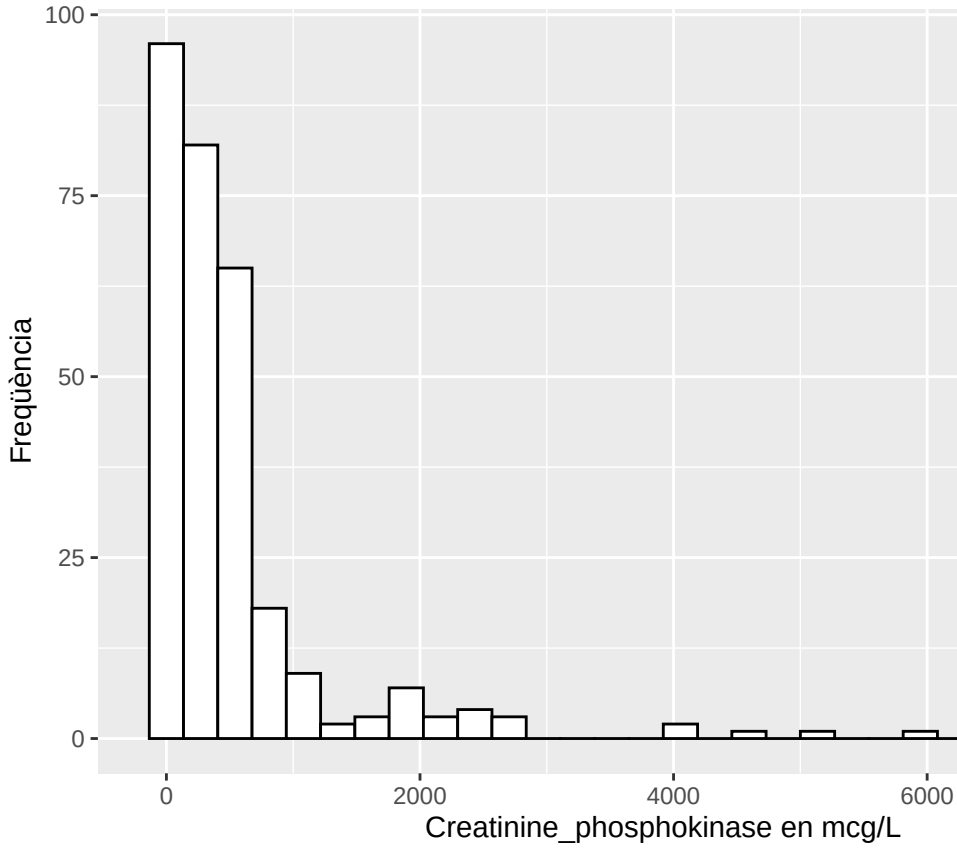
Doncs visualment sembla que la variable *age* sí que segueix una distribució normal. Però per a verificar-ho realitzarem un test d'hipòtesis. Primer mirem la mida de la mostra, com aquesta és superior a 50, hauríem d'aplicar el test de *Kolmogorov-Smirnov*. Però com desconeixem la μ i σ poblacionals, s'ha d'utilitzar el test de *Lilliefors* (Rodrigo 2016).

```
#Test de Lilliefors
library("nortest")
lillie.test(heart_data$age)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heart_data$age
## D = 0.069751, p-value = 0.001304
```

El test de Lilliefors ens dóna que el p-valor = 0.001304. Perquè segueixi normalitat amb un nivell de confiança del 95%, el p-valor ha de ser superior a 0.05. Com $0.001304 < 0.05$. Rebutgem la H_0 i diem que la variable *age* no segueix una distribució normal.

```
ggplot(heart_data, aes(x = creatinine_phosphokinase)) + geom_histogram(fill="white",colour="black") + yl
```



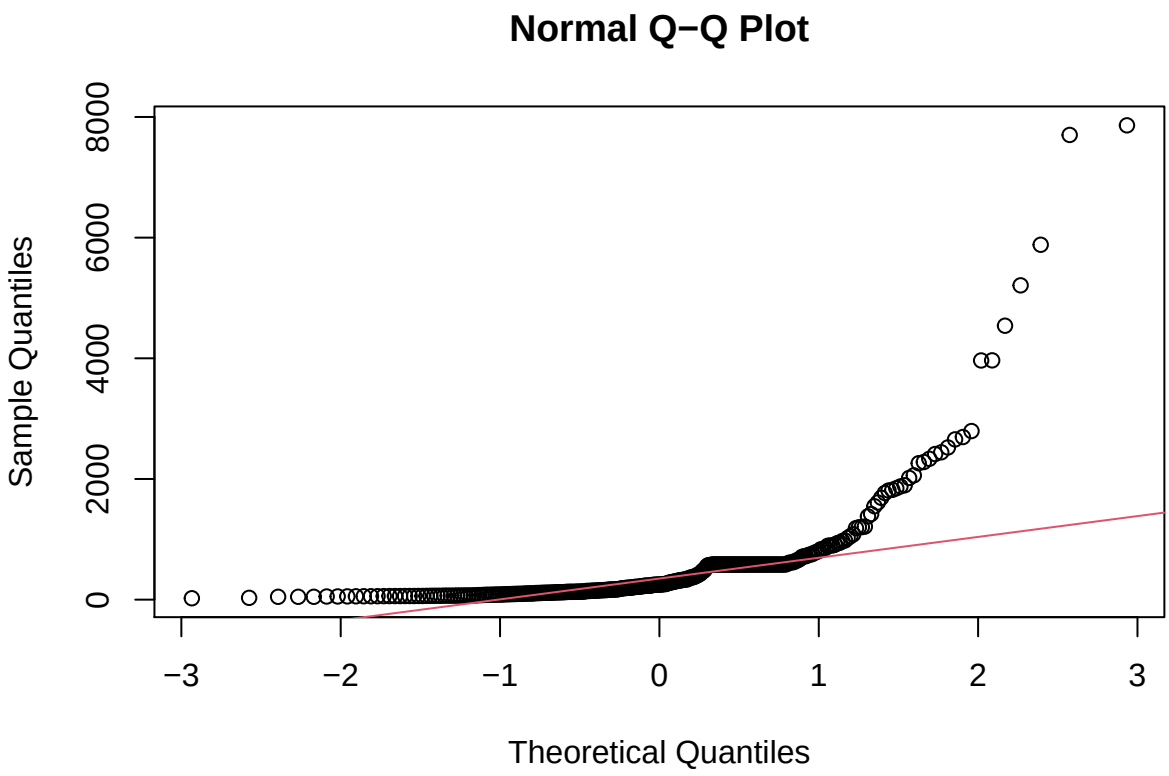
3.3.2.2 creatinine_phosphokinase

No sembla que segueixi una distribució normal. Definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

H_0 : La variable *creatinine_phosphokinase* segueix una distribució normal.

H_1 : La variable *creatinine_phosphokinase* no segueix una distribució normal.

```
#Observem la distribució amb la funció qqnorm
qqnorm(heart_data$creatinine_phosphokinase);qqline(heart_data$creatinine_phosphokinase, col = 2)
```



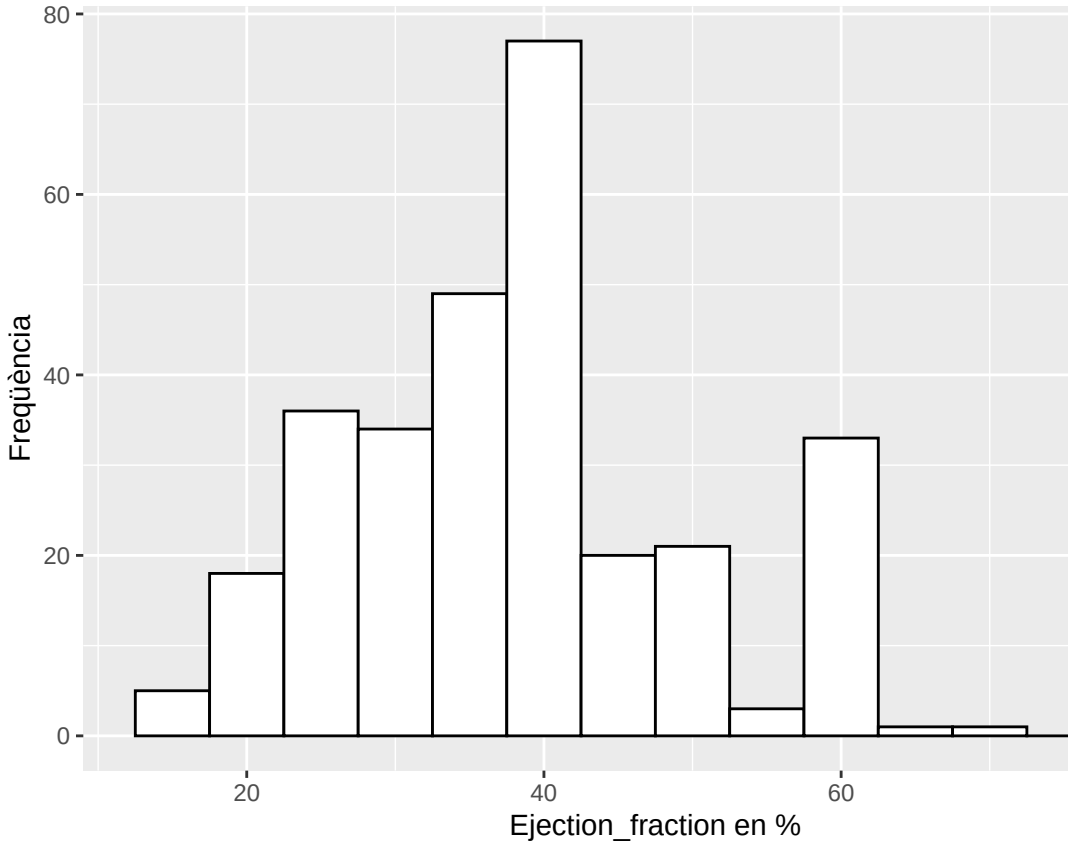
Veiem que s'allunya molt de com hauria de ser una distribució normal, de totes maneres ho comprobarem amb el test de *Lilliefors*.

```
#Test de Lilliefors
lillie.test(heart_data$creatinine_phosphokinase)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heart_data$creatinine_phosphokinase
## D = 0.28676, p-value < 2.2e-16
```

Efectivament, el p-valor és més petit que 0.05 i per tant rebutgem la H_0 i diem que la variable *creatinine_phosphokinase* no segueix una distribució normal.

```
breaks <- pretty(range(heart_data$ejection_fraction), n = nclass.FD(heart_data$ejection_fraction), min.
bwidth <- breaks[2]-breaks[1]
ggplot(heart_data, aes(x = ejection_fraction)) + geom_histogram(binwidth=bwidth,fill="white",colour="bl
```



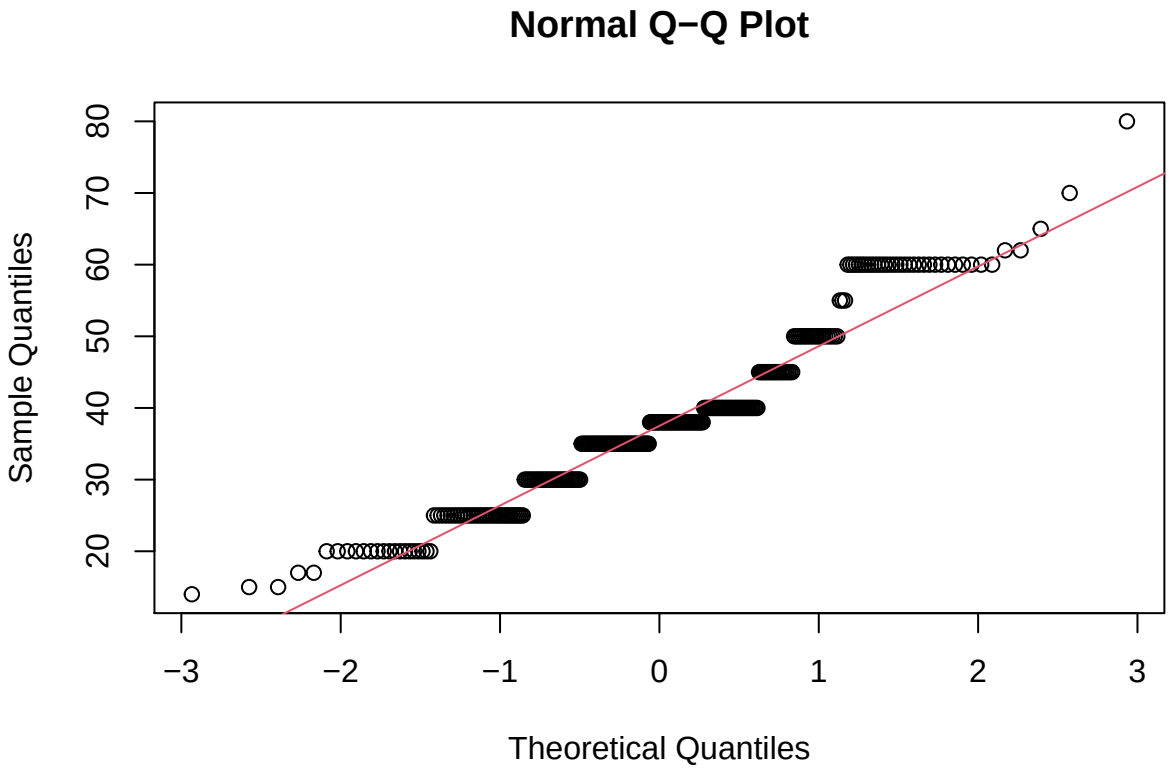
3.3.2.3 Ejection_fraction

L'inici del gràfic si que sembla que segueix una distribució normal però a partir de la mediana sembla que no, per tant com en els apartats anteriors realitzarem un test per sortir de dubtes. Definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

H_0 : La variable *ejection_fraction* segueix una distribució normal.

H_1 : La variable *ejection_fraction* no segueix una distribució normal.

```
#Observem la distribució amb la funció qqnorm
qqnorm(heart_data$ejection_fraction);qqline(heart_data$ejection_fraction, col = 2)
```



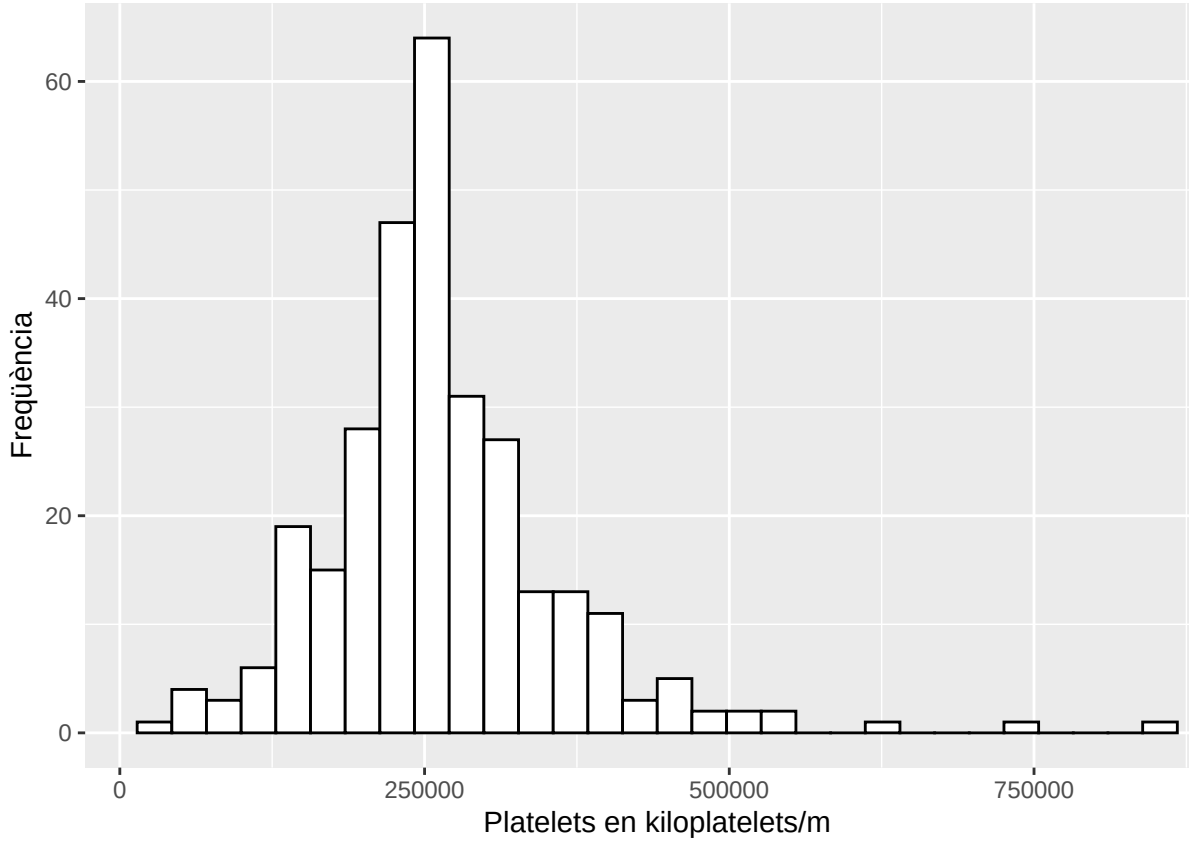
Veiem que les mostres més o menys van seguint la linea que representa una distribució normal. A continuació realitzarem el test de *Lilliefors* per acabar de verificar-ho.

```
#Test de Lilliefors
lillie.test(heart_data$ejection_fraction)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heart_data$ejection_fraction
## D = 0.16812, p-value < 2.2e-16
```

Doncs el p-valor és més petit que 0.05 i per tant rebutgem la H_0 i diem que la variable *ejection_fraction* no segueix una distribució normal.

```
ggplot(heart_data, aes(x = platelets)) + geom_histogram(fill="white",colour="black") + ylab("Freqüència
```



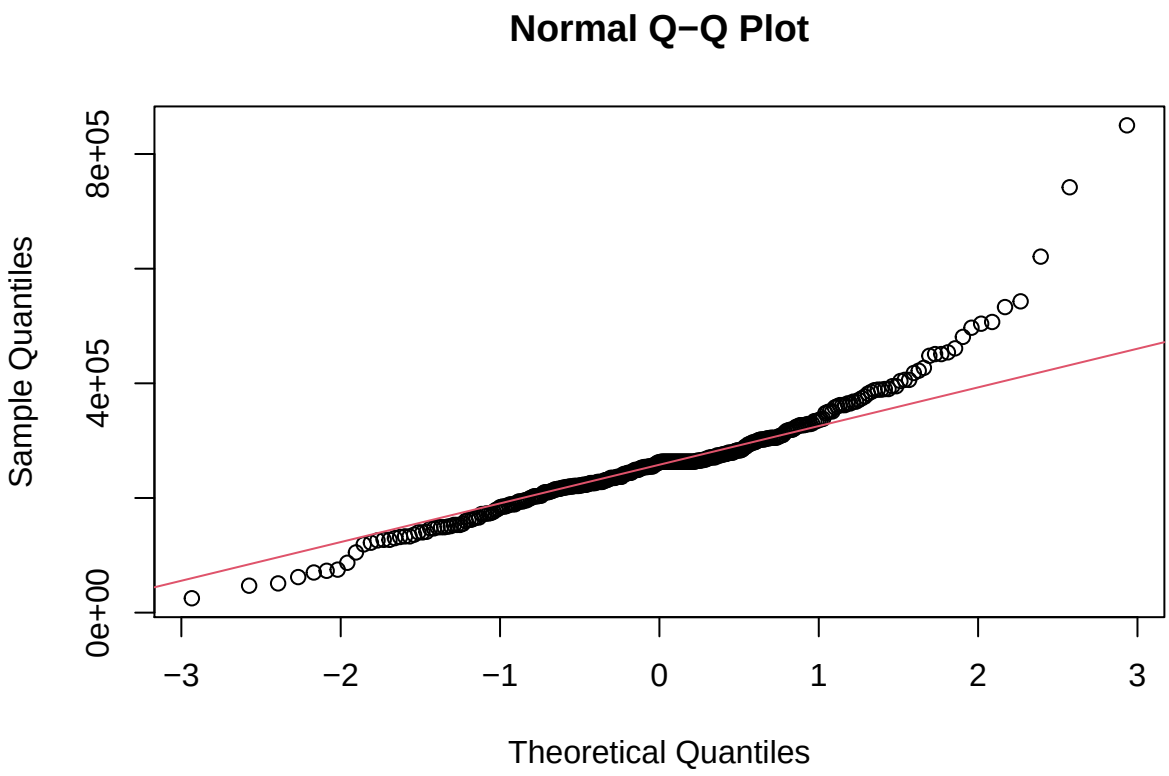
3.3.2.4 Platelets

La distribució de la variable *platelets* succeeix igual que en l'apartat anterior, no sembla que segueixi una distribució normal. A continuació ho verificarem definint les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

H_0 : La variable *platelets* segueix una distribució normal.

H_1 : La variable *platelets* no segueix una distribució normal.

```
#Observem la distribució amb la funció qqnorm
qqnorm(heart_data$platelets);qqline(heart_data$platelets, col = 2)
```



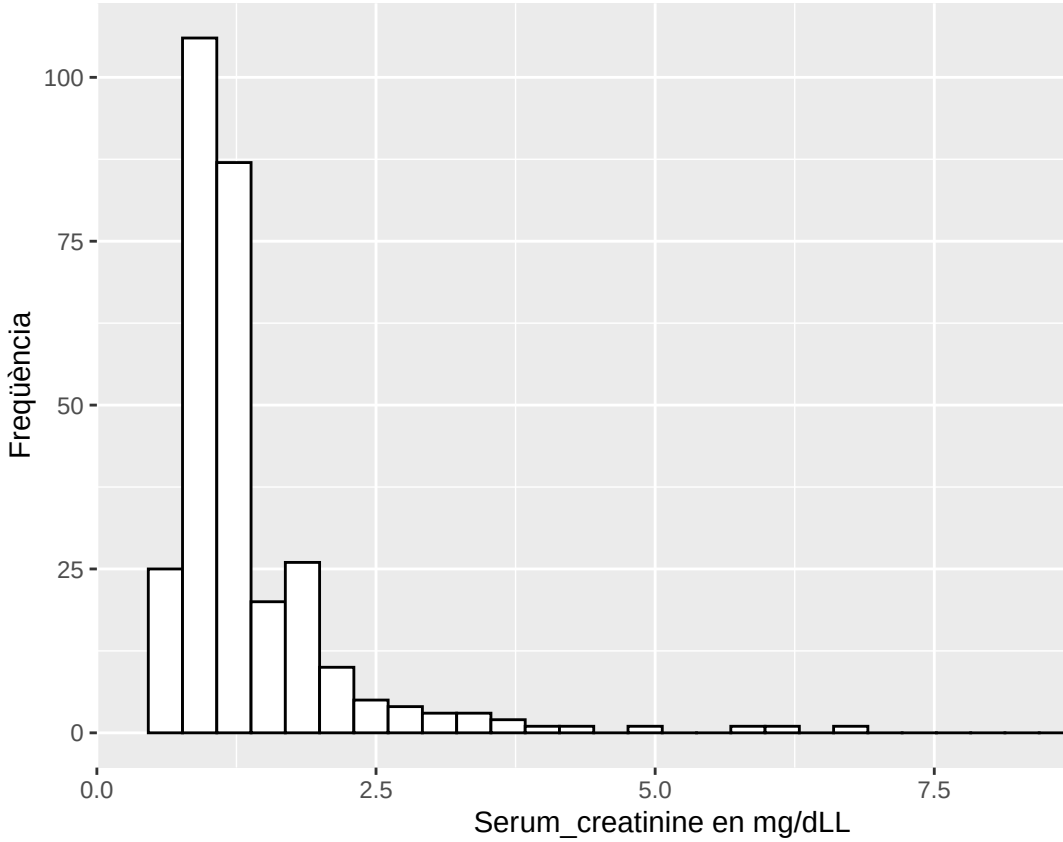
Veiem que les dades segueixen la línia que representa la distribució normal fins al quantil numero 1, a partir d'allà es desvia. Realitzarem el test de *Lilliefors* per acabar de verificar-ho.

```
#Test de Lilliefors
lillie.test(heart_data$platelets)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heart_data$platelets
## D = 0.11607, p-value = 1.904e-10
```

Doncs el p-valor (1.904e-10) és més petit que 0.05 i per tant rebutgem la H_0 i diem que la variable *platelets* no segueix una distribució normal.

```
ggplot(heart_data, aes(x = serum_creatinine)) + geom_histogram(fill="white",colour="black") + ylab("Fre
```



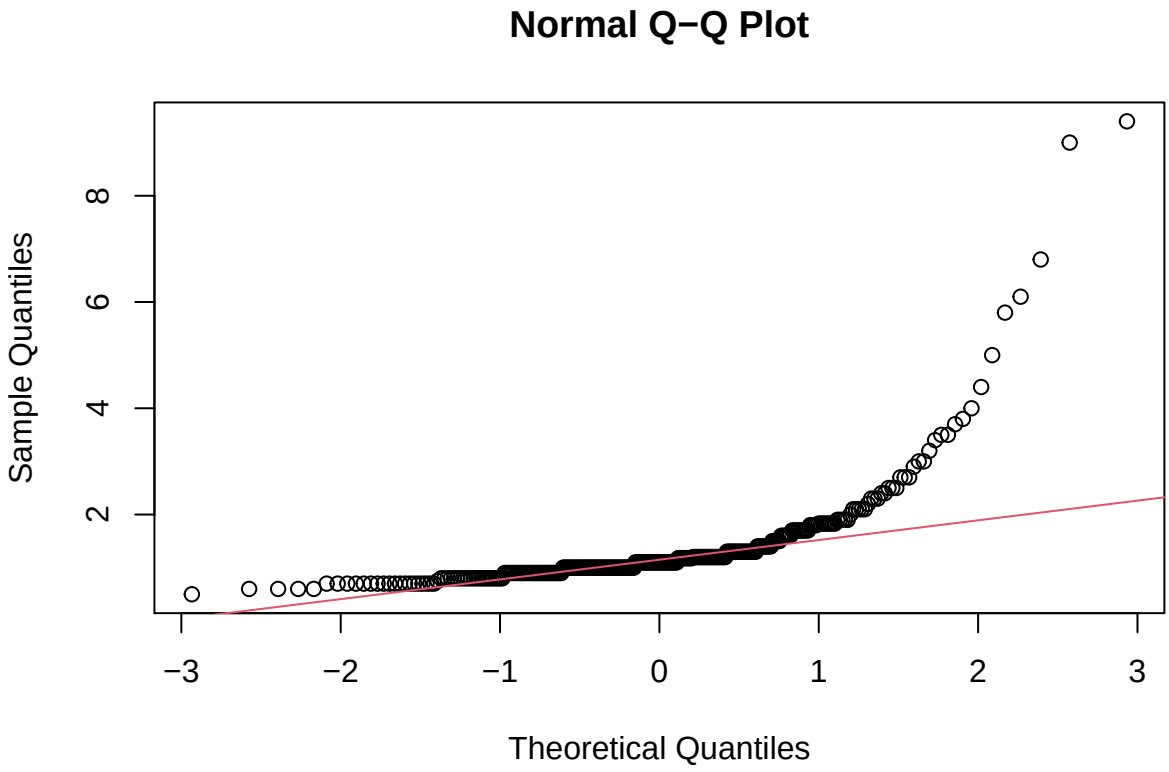
3.3.2.5 Serum_creatinine

Observem clarament com la variable *serum_creatinine* no té forma de Campana de Gauss i no segueix una distribució normal. De totes maneres ens n'assegurarem definint les hipòtesis nul · la i alternativa i avaluant si compleix l'assumpció de normalitat.

H_0 : La variable *serum_creatinine* segueix una distribució normal.

H_1 : La variable *serum_creatinine* no segueix una distribució normal.

```
#Observem la distribució amb la funció qqnorm
qqnorm(heart_data$serum_creatinine);qqline(heart_data$serum_creatinine, col = 2)
```



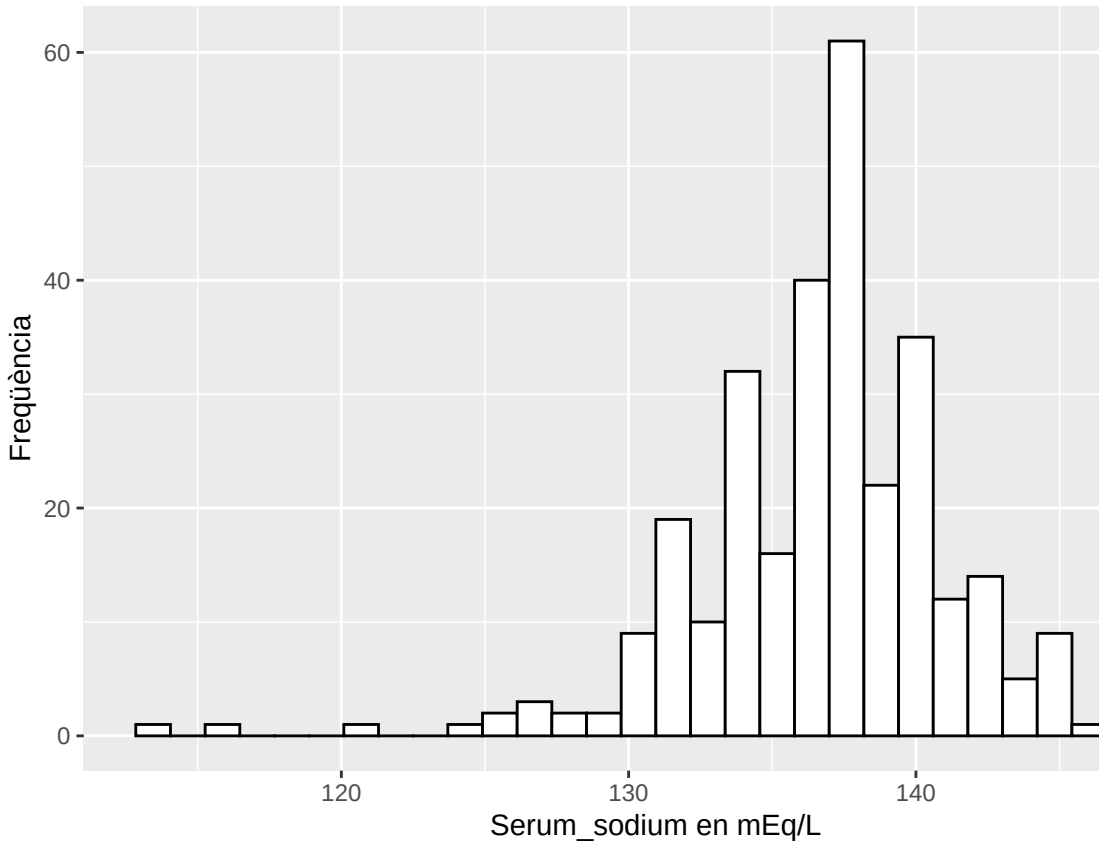
Veiem que com en la variable *platelets*, les dades segueixen la línia que representa la distribució normal fins al quantil numero 1, a partir d'allà es desvia. Realitzarem el test de *Lilliefors* per acabar de verificar-ho.

```
#Test de Lilliefors
lillie.test(heart_data$serum_creatinine)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heart_data$serum_creatinine
## D = 0.26525, p-value < 2.2e-16
```

Efectivament el p-valor (2.2e-16) és més petit que 0.05 i per tant rebutgem la H_0 i diem que la variable *serum_creatinine* no segueix una distribució normal.

```
ggplot(heart_data, aes(x = serum_sodium)) + geom_histogram(fill="white",colour="black") + ylab("Freqüèn
```



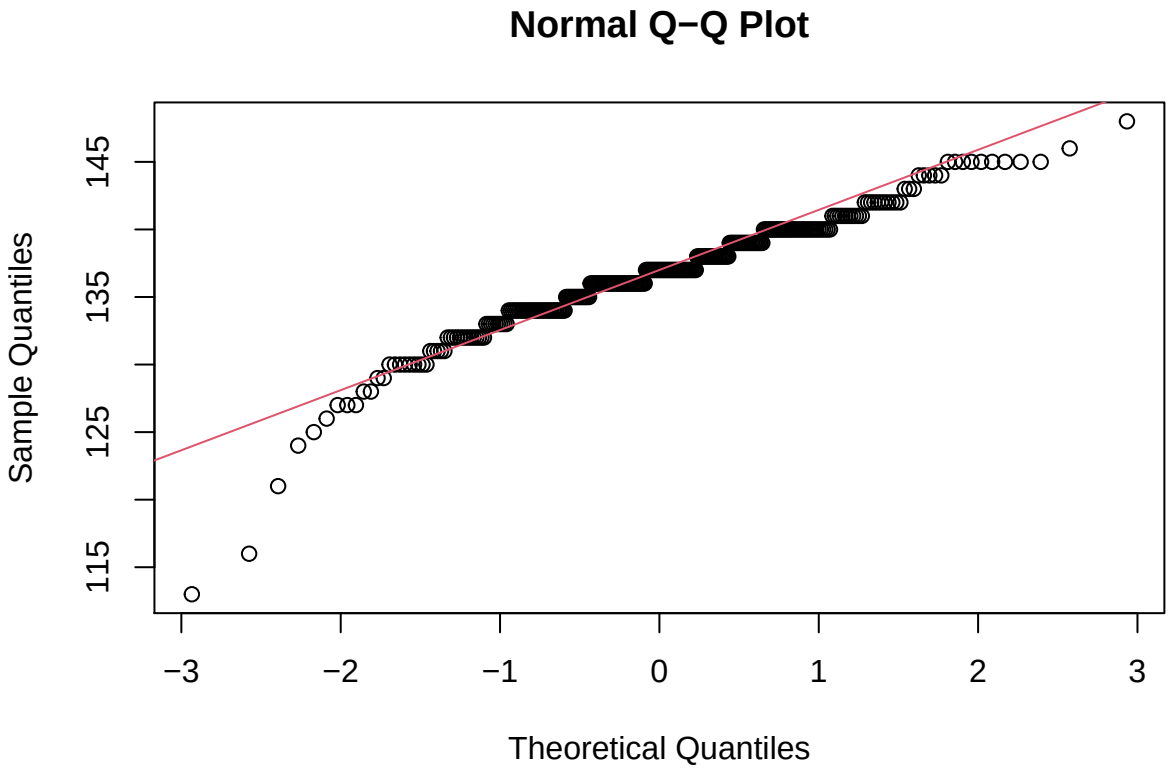
3.3.2.6 Serum_sodium

La distribució de la variable *serum_sodium* podria ser normal, per això ho comprovarem definint les hipòtesis nul · la i alternativa i avaluant si compleix l'assumpció de normalitat.

H_0 : La variable *serum_sodium* segueix una distribució normal.

H_1 : La variable *serum_sodium* no segueix una distribució normal.

```
#Observem la distribució amb la funció qqnorm
qqnorm(heart_data$serum_sodium);qqline(heart_data$serum_sodium, col = 2)
```



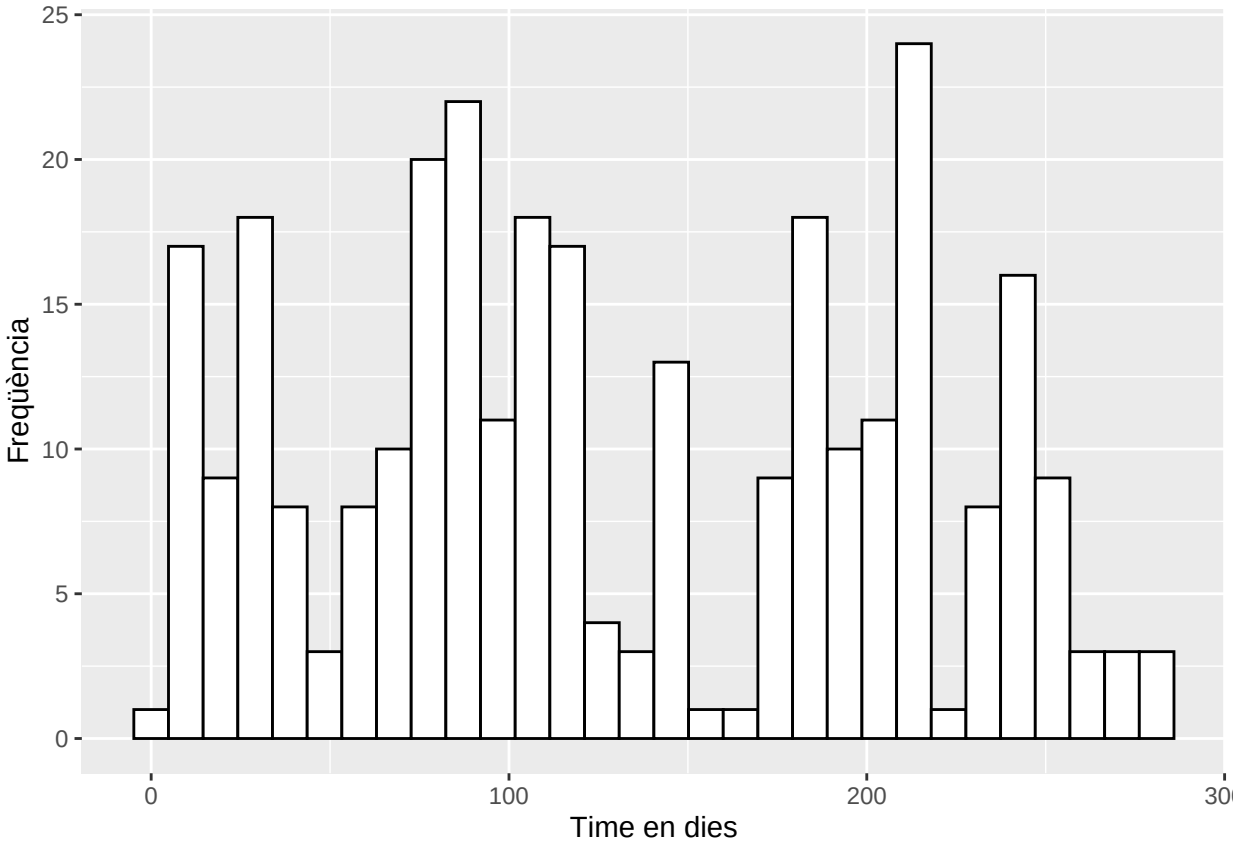
Observem que les dades segueixen bastant la línia que representa la distribució normal. A continuació realitzarem el test de *Lilliefors* per assegurar-nos si realment *serum_sodium* segueix una distribució normal amb un nivell de confiança del 95%.

```
#Test de Lilliefors
lillie.test(heart_data$serum_sodium)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heart_data$serum_sodium
## D = 0.11254, p-value = 8.683e-10
```

Doncs el p-valor = 8.683e-10, per tant com aquest és més petit que 0.05, rebutgem la H_0 i diem que la variable *serum_sodium* no segueix una distribució normal.

```
breaks <- pretty(range(heart_data$time), n = nclass.FD(heart_data$time), min.n = 1)
bwidth <- breaks[2]-breaks[1]
ggplot(heart_data, aes(x = time)) + geom_histogram(fill="white",colour="black") + ylab("Freqüència") + x
```



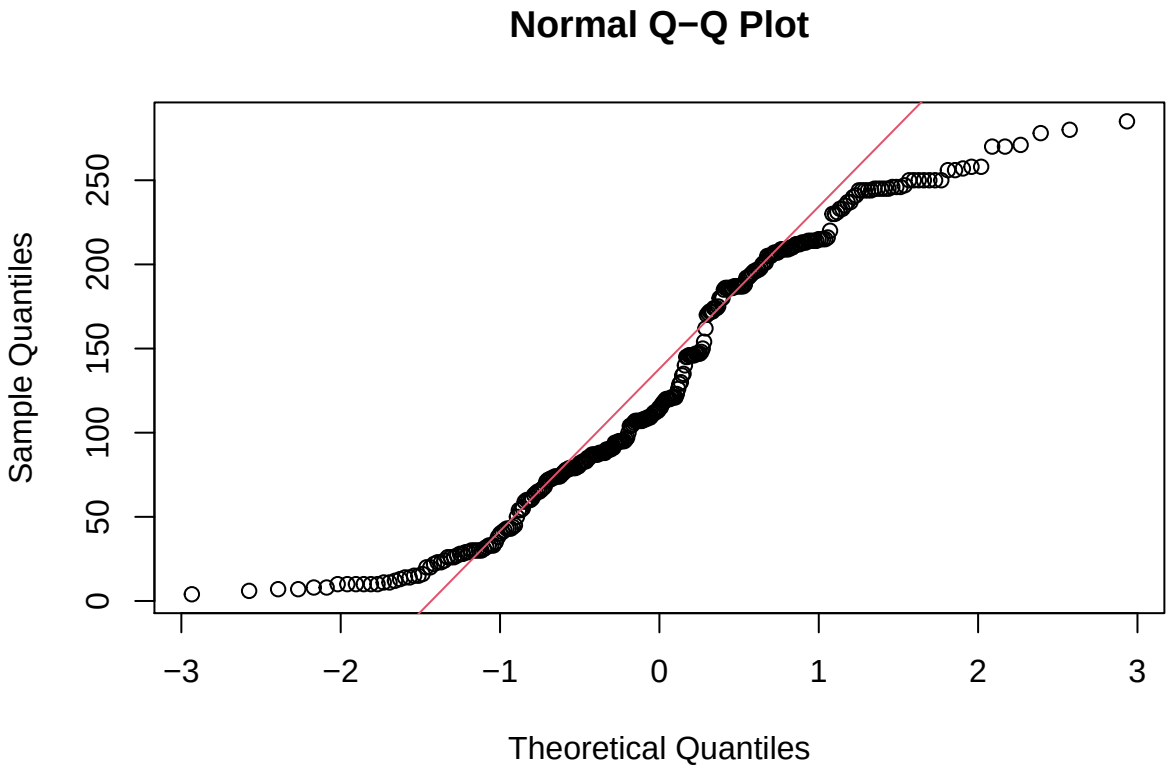
3.3.2.7 Time

La distribució de la variable *time* no s'assembla en res a una distribució normal que té la forma d'una Campana de Gauss, però de totes maneres definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

H_0 : La variable *time* segueix una distribució normal.

H_1 : La variable *time* no segueix una distribució normal.

```
#Observem la distribució amb la funció qqnorm
qqnorm(heart_data$time);qqline(heart_data$time, col = 2)
```




```
#Test de Lilliefors
lillie.test(heart_data$time)
```

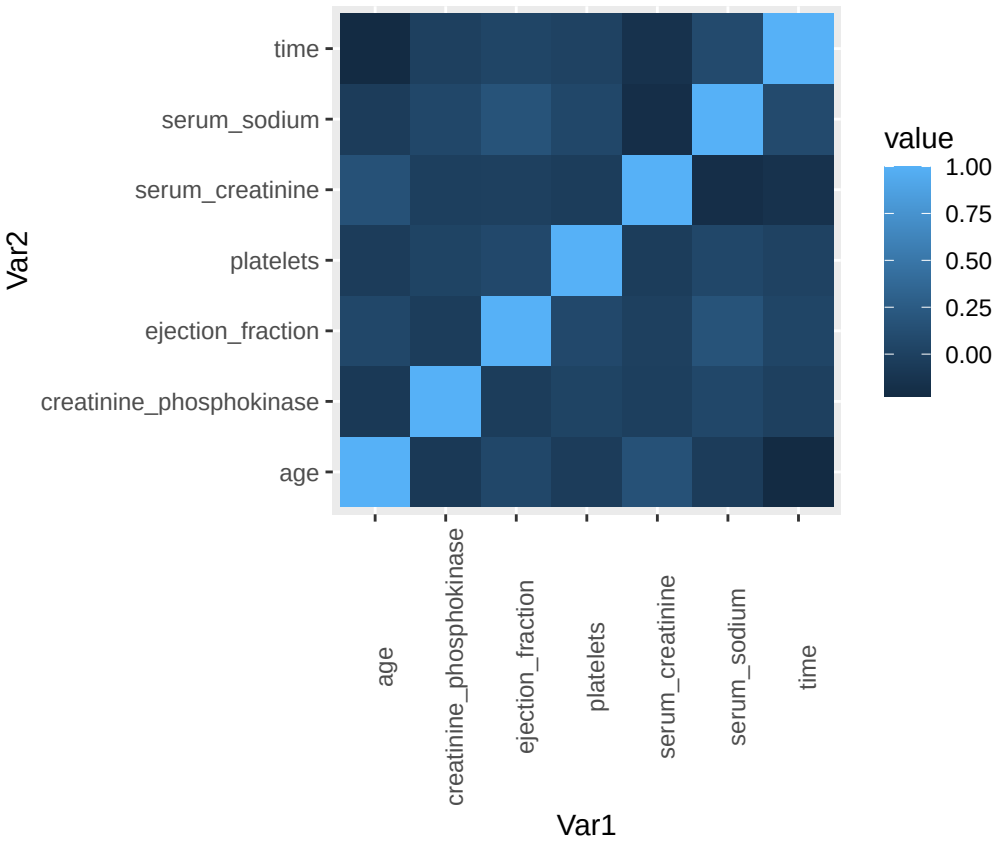
```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heart_data$time
## D = 0.10481, p-value = 2.01e-08
```

Efectivament en la representació de la *qqnorm*, les dades s'allunyen de la *qqline*. I el test de *Lilliefors* ens diu que el p-valor = 2.01e-08, per tant com aquest és més petit que 0.05, rebutgem la H_0 i diem que la variable *time* no segueix una distribució normal.

3.3.3 Correlacions

Per veure quins conjunts de variables estan relacionats entre si, farem servir l'anàlisi multivariant.

```
library(reshape2)
heat <- heart_data[, c('age', 'creatinine_phosphokinase', 'ejection_fraction', 'platelets',
                       'serum_creatinine', 'serum_sodium', 'time')]
qplot(x=Var1, y=Var2, data=melt(cor(heat, use="p")), fill=value, geom="tile") +
   theme(axis.text.x = element_text(angle = 90)) +
   coord_fixed()
```

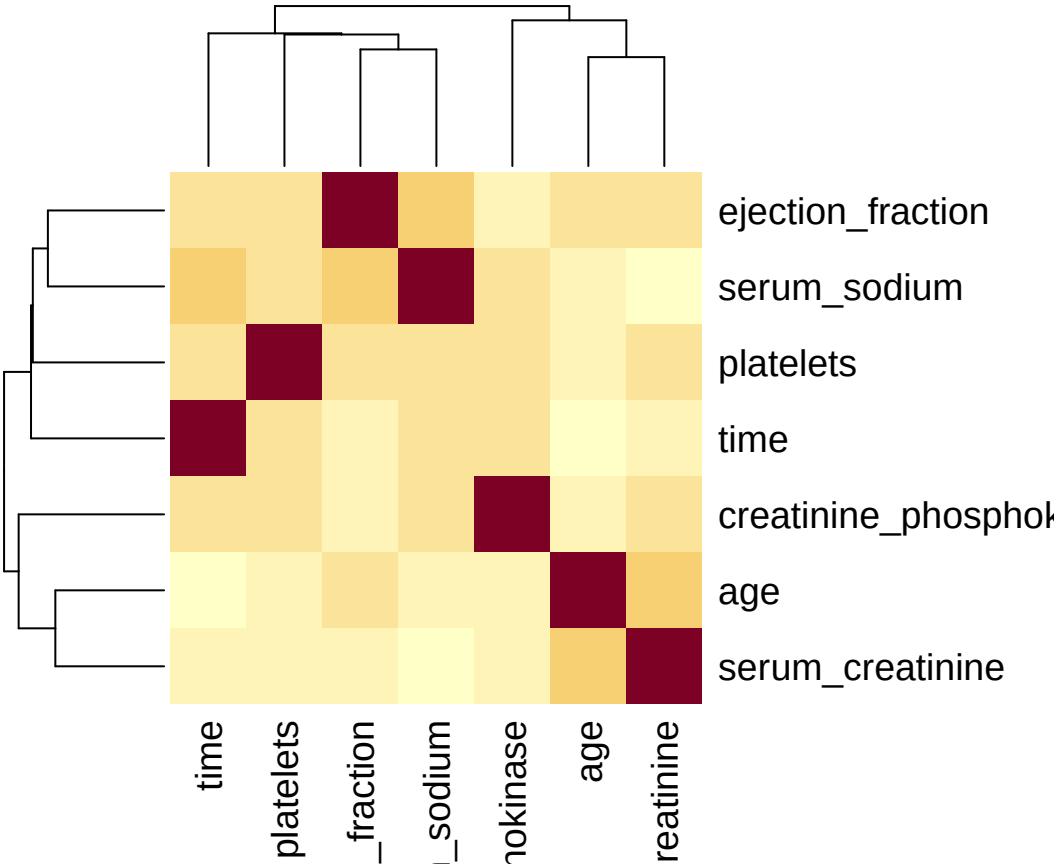


Com més intentistat de color, vol dir que més correlacionades estan les variables. Observem que les variables que tenen una correlació més alta són:

- serum_sodium - platelets
- time - age
- time - platelets
- time - serum_creatinine
- serum_sodium - serum_creatinine

Ara utilitzarem el *heatmap* per agrupar les variables que tenen més relació entre elles. Aquest utilitza un algorisme de *clustering jeràrquic* per a agrupar les variables.

```
trainscaled <- as.matrix(scale(cor(heat, use="p")))
heatmap(trainscaled, Colv=F, scale='none')
```



Podem observar en vermell més fort les variables que tenen més relació i com es van agrupant en un *clustering jeràrquic*.

3.4 Tasques de neteja i condicionat del joc de dades

3.4.1 Binarització de variables

Tornarem a passar les variables categòriques del joc de dades a binàries com teníem a l'inici de la pràctica perquè puguin ser usades en processos de modelatge perquè molts d'aquests procesoss no accepten variables categòriques.

```
#Realizem còpia
#anaemia
heart_data$anaemia <- heart_data1$anaemia
#diabetes
heart_data$diabetes <- heart_data1$diabetes
#high_blood_pressure
heart_data$high_blood_pressure <- heart_data1$high_blood_pressure
#sex
heart_data$sex <- heart_data1$sex
#smoking
heart_data$smoking <- heart_data1$smoking
#DEATH_EVENT
heart_data$DEATH_EVENT <- heart_data1$DEATH_EVENT
#Comprovem que les transformacions s'hagin realitzat correctament
str(heart_data)
```

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

3.5 Discretització

Veient les diferents variables que tenim, crec que només té sentit discretitzar la variable *ejection_fraction*, realitzarem una discretització on separarem cada 20%, és a dir 0-20%, 20-40%, 40-60%, 60-80% i 80-100%. I mirarem com varia la distribució de les dades.

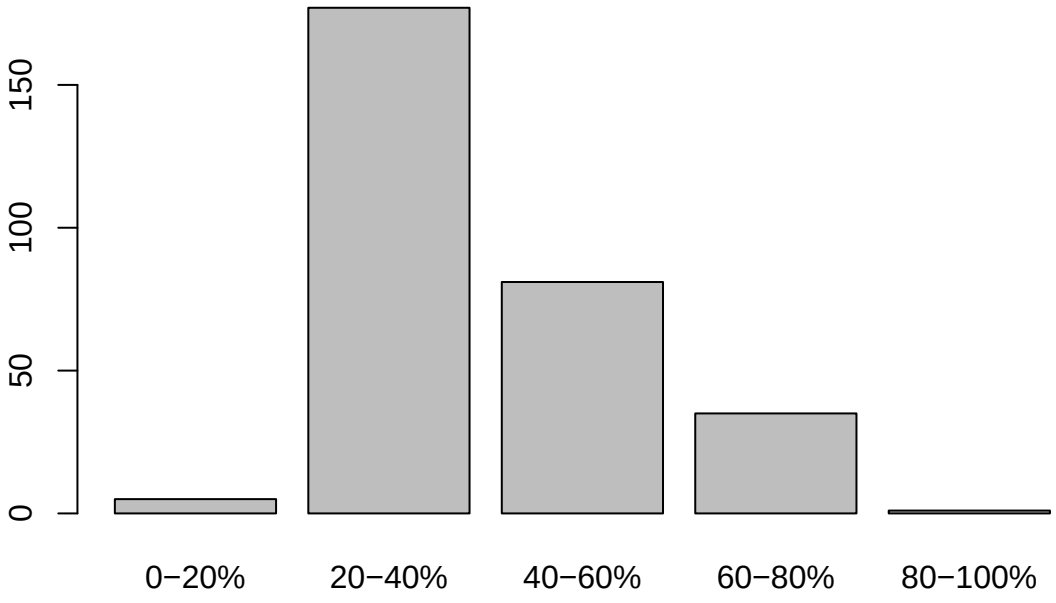
```
#Discretitzem ejection_fraction
heart_data$ejection_fraction_discr[heart_data$ejection_fraction < 20] <- "0-20%"
heart_data$ejection_fraction_discr[heart_data$ejection_fraction >= 20 & heart_data$ejection_fraction < 4
heart_data$ejection_fraction_discr[heart_data$ejection_fraction >= 40 & heart_data$ejection_fraction < 6
heart_data$ejection_fraction_discr[heart_data$ejection_fraction >= 60 & heart_data$ejection_fraction < 8
heart_data$ejection_fraction_discr[heart_data$ejection_fraction >= 80] <- "80-100%"
```

Normalment eliminariem la columna discretitzada però en aquest cas la mantindrem perquè en funció del modelatge que volguem realitzar en un futur, podriem necessitar la variable en format numèric.

```
head(heart_data)
```

```
##   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 ejection_fraction_discr
## 1           1                 20-40%
## 2           1                 20-40%
## 3           1                 20-40%
## 4           1                 20-40%
## 5           1                 20-40%
## 6           1                 40-60%
```

```
plot(as.factor(heart_data$ejection_fraction_discr))
```



3.6 Estudi PCA

L'anàlisis de components principals o PCA (*principal component analysis*) és una tècnica que s'utilitza per descriure un conjunt de dades amb noves variables no correlacionades. Una de les aplicacions del PCA és la reducció de la dimensionalitat d'un conjunt de dades perdent la menor cantitat d'informació posible. ("Análisis de Componentes Principales," n.d.)

Quan tenim un gran nombre de variables cuantitatives posiblement correlacionades, permet reduir-les a un número menor de components principals o variables transformades que expliquen gran part de la variabilitat de les dades. També podem utilitzar-lo com a eina per a la visualització de les dades. (Martínez 2018)

Per començar amb l'anàlisis de components, hem d'estandaritzar totes les dades per a que segueixin una distribució normal. En l'anàlisis exploratòri hem observar que cap d'elles ho era. Per tant a continuació normalitzarem totes les variables.

Utilitzarem la funció *prcomp()*, aquesta ens centra les variables perquè tinguin una mitja de 0 i amb l'argument scale = TRUE, ens les escala amb una desviació estàndard d'1.

```
#Executem anàlisis pca
pca_data <- prcomp(heart_data1, scale = TRUE)
names(pca_data)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
#Mostra dels primers 6 elements del vector de loadings dels primers 4 components principals
head(pca_data$rotation)[,1:4]
```

```
##                                PC1         PC2        PC3         PC4
## age                     -0.34945554  0.02854017 -0.3426476  0.01034420
## anaemia                 -0.14388283  0.27218053 -0.3058006  0.34813693
## creatinine_phosphokinase  0.03329930 -0.19009778  0.2846400 -0.51254659
## diabetes                  0.07252016  0.26244240  0.4353483 -0.02525959
## ejection_fraction         0.20650121  0.24176217 -0.4085698 -0.08414023
## high_blood_pressure      -0.14954158  0.22897096 -0.2411413 -0.35200154
```

```
#Observem la dimensió
dim(pca_data$rotation)
```

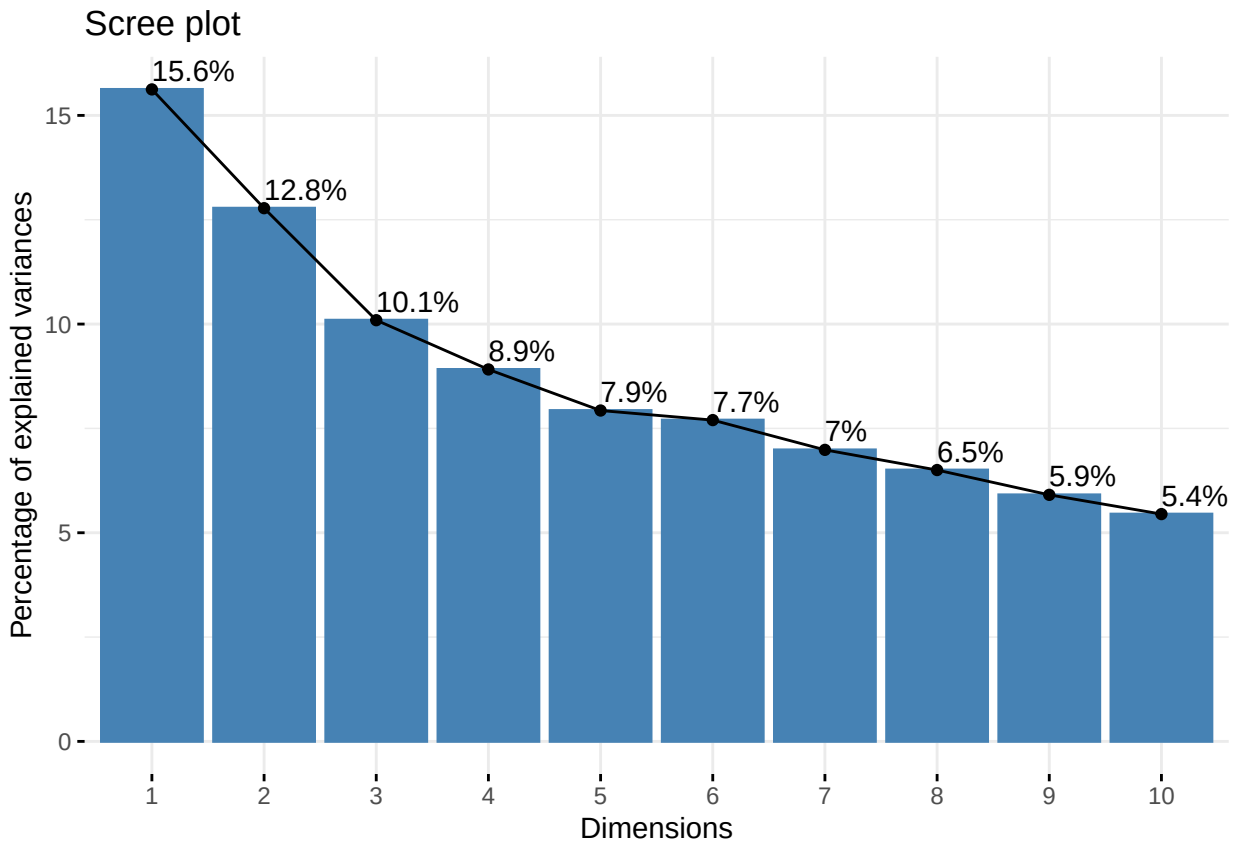
```
## [1] 13 13
```

```r
#Observem summary
summary(pca_data)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.4251 1.2888 1.1455 1.07640 1.01523 1.00044 0.95297
## Proportion of Variance 0.1562 0.1278 0.1009 0.08913 0.07928 0.07699 0.06986
## Cumulative Proportion  0.1562 0.2840 0.3849 0.47405 0.55333 0.63032 0.70018
##                           PC8     PC9    PC10    PC11    PC12    PC13
## Standard deviation     0.91940 0.87637 0.84149 0.82827 0.71666 0.61367
## Proportion of Variance 0.06502 0.05908 0.05447 0.05277 0.03951 0.02897
## Cumulative Proportion  0.76520 0.82428 0.87875 0.93152 0.97103 1.00000
```

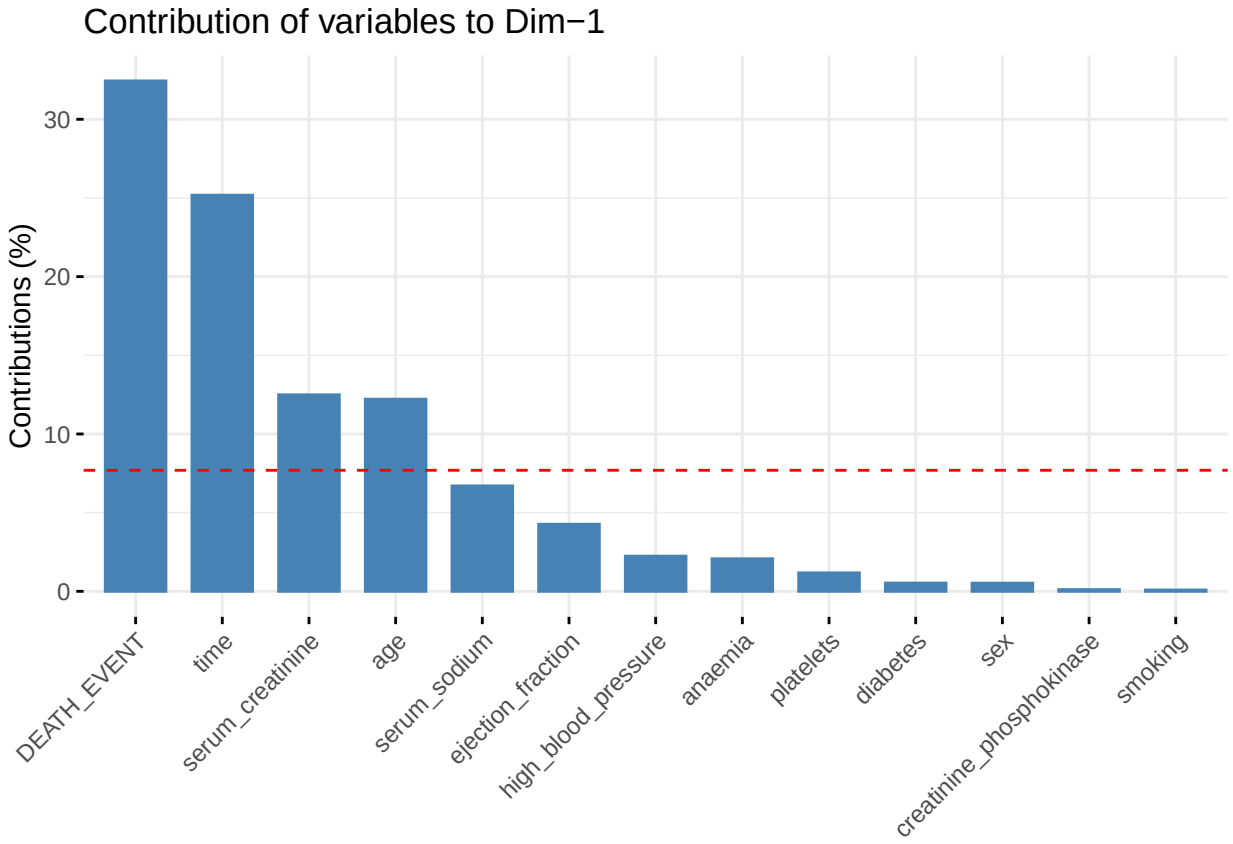
A continuació generament un *scree plot* que representi els eigenvalors ordenats de major a menor per a poder escollir quin és el numero òptim de components principals.

```r
library(factoextra)
#eigenvalues(screeplot)
fviz_screeplot(pca_data, addlabels = TRUE)
```



Apreciem que la diferència entre 1 dimensió i 4 dimensions és molt gran, però a partir d'aquí, entre 5 i 8 dimensions el pendent de la corva s'estabilitza bastant.

```r
#Contribució de les variables
fviz_contrib(pca_data, choice = "var", axes = 1)
```



En aquesta representació observem la contribució de les variables a una component. Si disposesim de moltes variables, podriem ometre les que menys contribueixen, com *smoking*, *creatinine_phosphokinase*, *sex* i *diabetes* però com només en tenim 13, nosaltres les tindrem totes en compte.

Ara calcularem la proporció de la variança explicada (PVE) per a cada component principal.

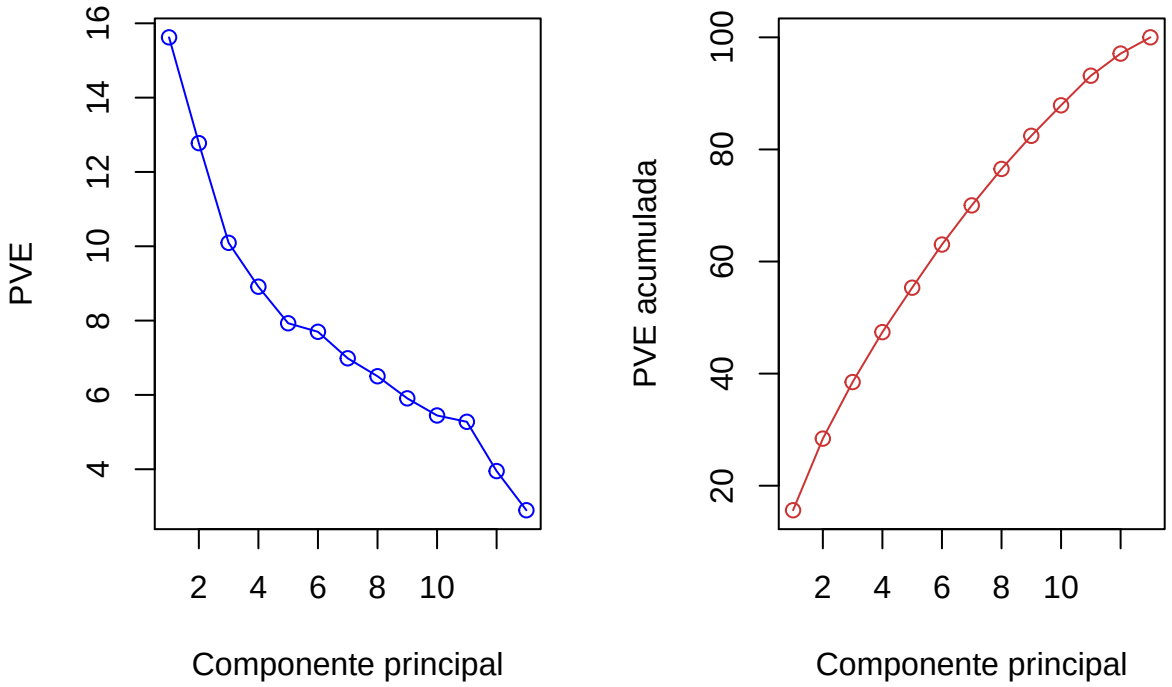
```
#Contribució de les variables en %
PVE <- 100*pca_data$sdev^2/sum(pca_data$sdev^2)
PVE
```

```
##  [1] 15.622596 12.776378 10.093015  8.912620  7.928468  7.699103  6.985781
##  [8]  6.502326  5.907925  5.446994  5.277144  3.950762  2.896889
```

Aquestes dades ens confirmen el que observàvem al gràfic dels *eigenvalues*, la primera component explica el 15.62% de la variança, la segona el 12.77%, mentre que per exemple la 13 dimensió només el 2.89%.

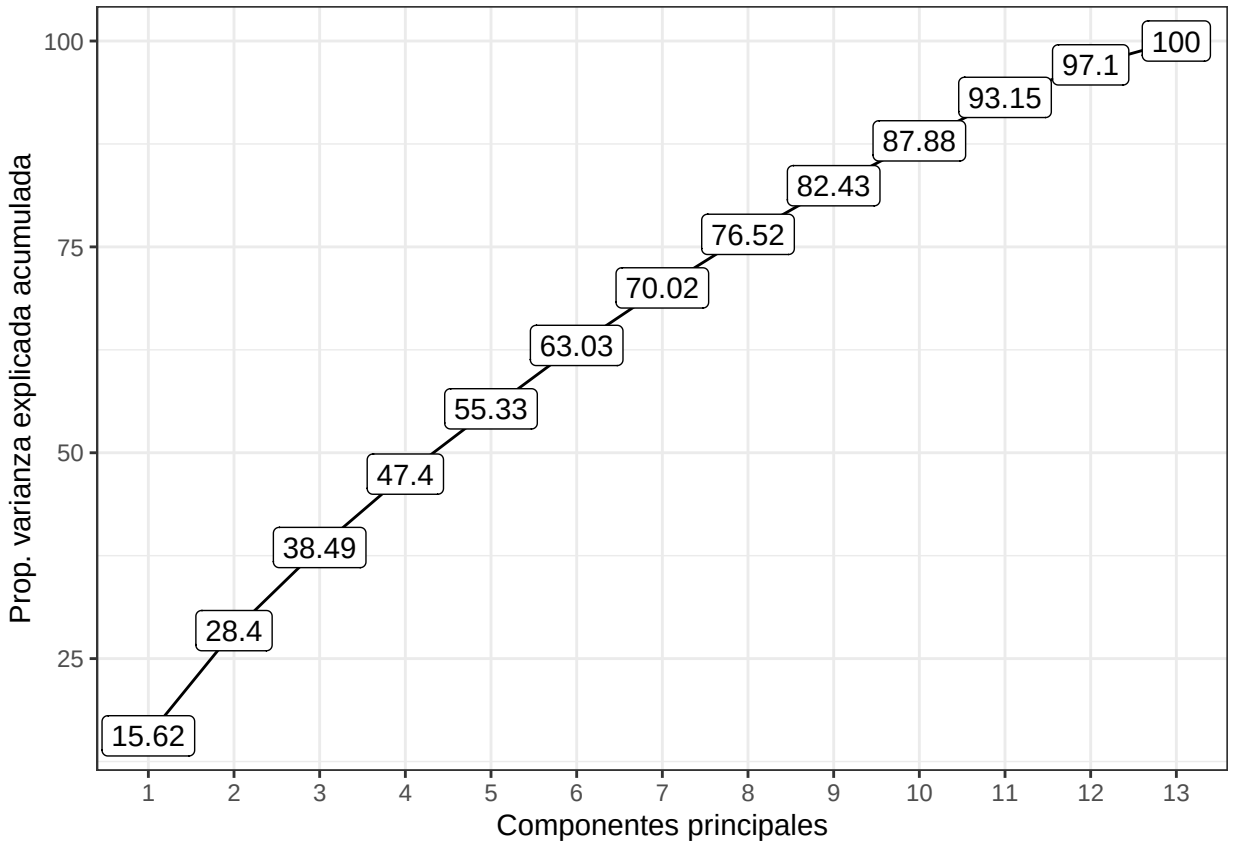
```
par(mfrow = c(1,2))

plot(PVE, type = "o",
     ylab = "PVE",
     xlab = "Componente principal",
     col = "blue")
plot(cumsum(PVE), type = "o",
     ylab = "PVE acumulada",
     xlab = "Componente principal",
     col = "brown3")
```



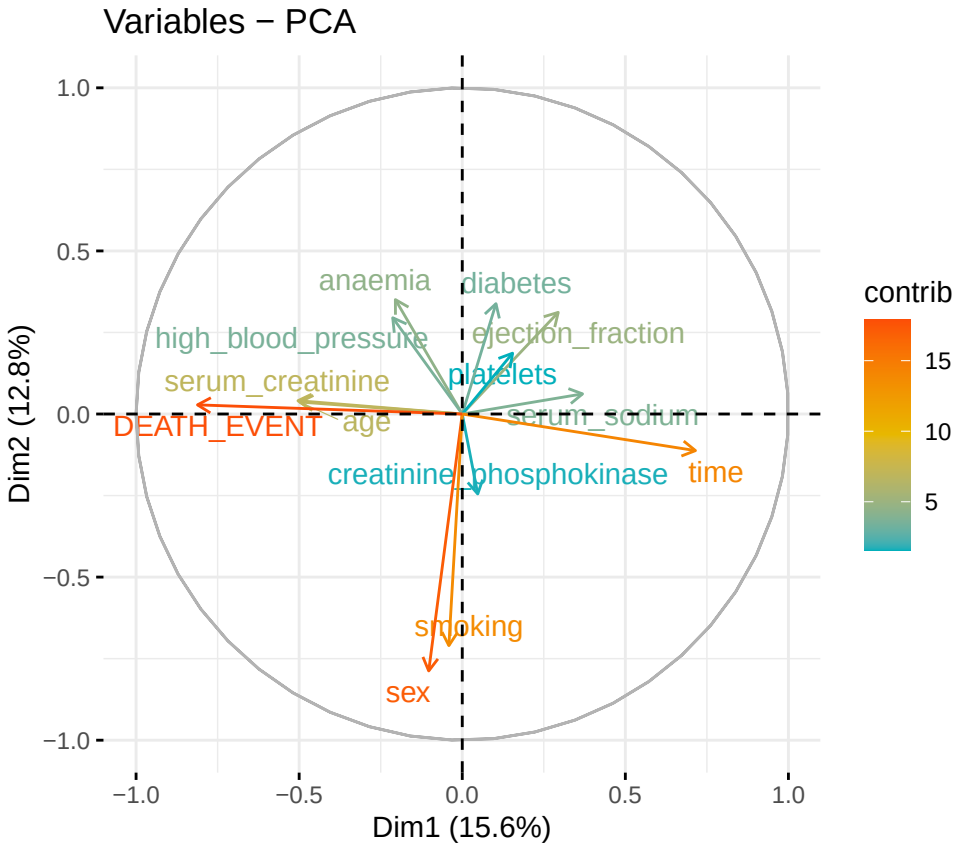
Els 7 primers components principals representen al voltant del 70% de la variança.

```
#Representació de la variança acumulada
prop_varianza_acum <- cumsum(PVE)
ggplot(data = data.frame(prop_varianza_acum, pc = factor(1:13)),
       aes(x = pc, y = prop_varianza_acum, group = 1)) +
  geom_point() +
  geom_line() +
  geom_label(aes(label = round(prop_varianza_acum,2))) +
  theme_bw() +
  labs(x = "Componentes principales",
       y = "Prop. varianza explicada acumulada")
```



En aquest gràfic encara ho veiem més clar, els primers 7 components principals, representen el 70.02% de la variança. Es veu clar que el pendent de la corba és molt constant, cosa que provoca que si volem reduïr la dimensionalitat, hem de perdre una part gairebé proporcional de la informació.

```
#Gràfic de variables.
fviz_pca_var(pca_data,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
)
```



Les variables que estan correlacionades apunten cap a la mateixa direcció, en canvi les que no ho estan apunten en direccions oposades. En aquest gràfic també observem que en funció de la contribució, les variables tenen una tonalitat amb tons més càlids.

Referències

"Análisis de Componentes Principales." n.d. https://es.wikipedia.org/wiki/An.

Martínez, Cristina Gil. 2018. "ANÁLISIS de Componentes Principales (Pca)."

Organización Mundial de la Salud. 2018. "Las 10 principales causas de defunción." https://www.who.int/es/news-room/fact-sheets/detail/the-top-10-causes-of-death.

Rodrigo, Joaquín Amat. 2016. "Análisis de Normalidad: Gráficos Y Contrastes de Hipótesiso."