Pràctica 2

Pol Moya Betriu i Xavier Martin Bravo

2 de enero, 2021

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%), cada any moren aproximadament 17.9 milions de persones. 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]*****:** descriu edat del pacient (anys).
- **anaemia** ***[factor]*****:** que especifica si el pacient pateix anèmia o no.
- **creatinine_phosphokinase** ***[integer]*****:** representa el nivell de *CPK* en la sang en (*mcg/L*).
- **diabetes** ***[factor]*****:** indica si el pacient és diabètic o no.
- **ejection_fraction** ***[integer]*****:** descriu el percentatge de sang que surt del cor en cada contracció en (%).
- **high_blood_pressure** ***[factor]*****:** indica si el pacient té hipertensió o no.
- **platelets** ***[numeric]*****:** representa el nombre de plaquetes en sang del pacient (kiloplatelets/mL).
- **serum_creatinine** ***[numeric]*****:** representa el nivell de creatinina en la sang (mg/dL).
- **serum_sodium** ***[integer]*****:** indica el nivell de sodi en sang (mEq/L).
- **sex** ***[factor]*****:** indica si el sexe del pacient és masculí o femení.
- **smoking** ***[factor]*****:** indica si el pacient fuma o no.
- **time** ***[integer]*****:** període de seguiment en dies.
- **DEATH_EVENT** ***[factor]*****:** 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=",", stringsAsFactors = FALSE)
#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.

```
#Expliquem amb l'atribut [anaemia] com fem la transformació de tipus.
#Obtenim un vector que conté les posicions on l'observació de l'atribut [anaemia] = 1
i <- heart_data$anaemia == '1'
#Hem decidit associar el valor '1' amb un 'Si'
heart_data$anaemia[i] <- "Si"
#Obtenim un vector que conté les posicions on l'observació de l'atribut [anaemia] = 0
i <- heart_data$anaemia == '0'
#Hem decidit associar el valor '0' amb un 'No'
heart_data$anaemia[i] <- "No"
#Finalment asignem aquest 2 valors 'Si'/'No' i convertim l'atribut a tipus categoric.
heart_data$anaemia <- as.factor(heart_data$anaemia)
#diabetes
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:

```
#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
```

Ara passarem a validar la tendència dels atributs numèrics per veure si estem davant de distribucions de dades simètriques o sesgades. Per aconseguir-ho definirem una funció que ens permetrà obtenir la *"moda"*, que és l'única messura de tendència central que la funció *"summary()"* no ens ha mostrat.

```
# Funció que ens retorna la "moda" d'un vector de dades
# Param:
#   v -> vector de dades a analitzar
# Return:
#   La "moda" del vector analitzat
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

#Obtenim la "moda" de l'atribut [age]
result <- getmode(heart_data$age)
print(paste("La moda per l'atribut [age] es: ",result))
```

```
## [1] "La moda per l'atribut [age] es:  60"
```

```r
#Obtenim la "moda" de l'atribut [creatinine_phosphokinase]
result <- getmode(heart_data$creatinine_phosphokinase)
print(paste("La moda per l'atribut [creatinine_phosphokinase] es: ",result))
```

```
## [1] "La moda per l'atribut [creatinine_phosphokinase] es:  582"
```

```r
#Obtenim la "moda" de l'atribut [ejection_fraction]
result <- getmode(heart_data$ejection_fraction)
print(paste("La moda per l'atribut [ejection_fraction] es: ",result))
```

```
## [1] "La moda per l'atribut [ejection_fraction] es:  35"
```

```r
#Obtenim la "moda" de l'atribut [platelets]
result <- getmode(heart_data$platelets)
print(paste("La moda per l'atribut [platelets] es: ",result))
```

```
## [1] "La moda per l'atribut [platelets] es:  263358.03"
```

```r
#Obtenim la "moda" de l'atribut [serum_creatinine]
result <- getmode(heart_data$serum_creatinine)
print(paste("La moda per l'atribut [serum_creatinine] es: ",result))
```

```
## [1] "La moda per l'atribut [serum_creatinine] es:  1"
```

```r
#Obtenim la "moda" de l'atribut [serum_sodium]
result <- getmode(heart_data$serum_sodium)
print(paste("La moda per l'atribut [serum_sodium] es: ",result))
```

```
## [1] "La moda per l'atribut [serum_sodium] es:  136"
```

```r
#Obtenim la "moda" de l'atribut [time]
result <- getmode(heart_data$time)
print(paste("La moda per l'atribut [time] es: ",result))
```

```
## [1] "La moda per l'atribut [time] es:  187"
```

Si ara agafem totes les mesures de tendencia central d'aquests atributs podrem fer-nos una idea de la tendència que tenen.

L'atribut **age**

- Mean: 60.83
- Median: 60.00
- Mode: 60.00

Les 3 messures són pràcticament iguals i per tant la distribució de les dades serà simètrica.

L'atribut **creatinine_phosphokinase**

- Mean: 581.00
- Median: 250.00
- Mode: 582.00

Hi han 2 messures molt iguals i una de molt diferent i per tant la distribució de les dades serà sesgada.

L'atribut **ejection_fraction**

- Mean: 38.08
- Median: 38.00
- Mode: 35.00

Les 3 messures són pràcticament iguals i per tant la distribució de els dades serà simètrica.

L'atribut **platelets**

- Mean: 263358.00
- Median: 262000.00
- Mode: 263358.03

Hi ha 2 messures iguals i una altre una mica diferent i per tant la distribució de les dades serà sesgada.

L'atribut **serum_creatinine**

- Mean: 1.394
- Median: 1.100
- Mode: 1.000

Hi han 2 messures pràcticament iguals i una altre diferent i per tant la distribució de les dades serà sesgada.

L'atribut **serum_sodium**

- Mean: 136.6
- Median: 137.0
- Mode: 136.0

Les 3 messures són pràcticament iguals i per tant la distribució de les dades serà simètrica.

L'atribut **time**

- Mean: 130.3
- Median: 115.0
- Mode: 187.0

Les 3 messures són diferents i per tant la distribució de les dades serà sesgada.

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 nostre joc de dades per tant no ens és necessari tractar-los i podem seguir endavant. Però si en disposéssim, tindríem diverses opcions per a tractar-los, nosaltres ens decantaríem per si la variable té un percentatge molt elevat de valors no disponibles, eliminaríem directament la variable. En canvi si el percentatge de valors no disponibles és baix, el que realitzaríem és reemplaçar aquests per la mediana.

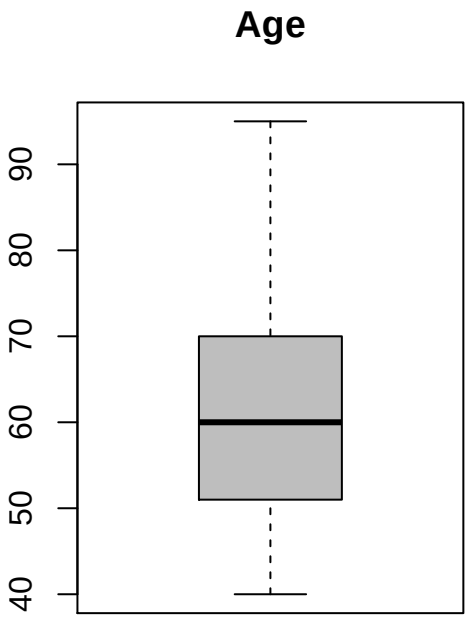
3.3 Identificació i tractament de valors extrems.

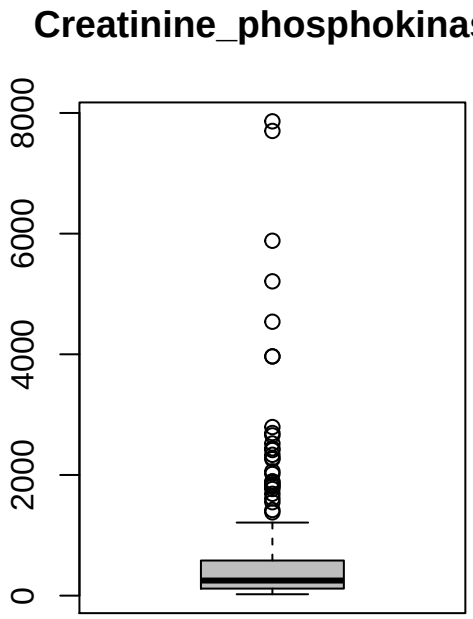
Visualitzarem els valors extrems o *outliers* amb un *boxplot* per cada variable numèrica.

Els gràfics tipus *boxplot* mostra 5 valors de la distribució de les observacions de l'atribut avaluat:

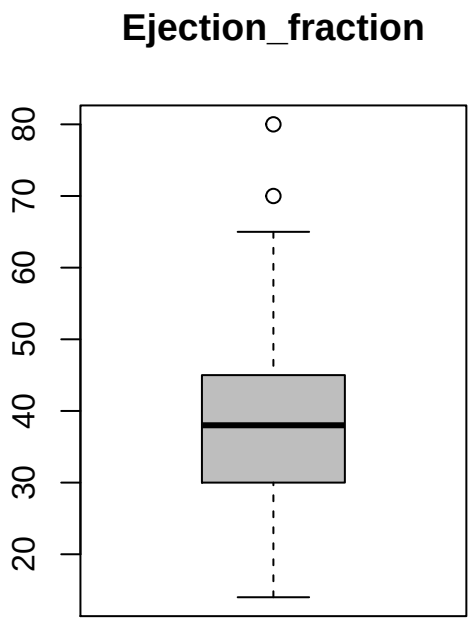
- Extrem superior de la capsa indica el *3º Quartil*.
- Extrem inferior de la capsa indica el *1º Quartil*.
- La *"median"* està reflectida amb una ratlla dins de la capsa.
- Les 2 línies *"whiskers"* ubicades fora de la capsa indiquen el valor mínim i màxim. El màxim = (3º Quartil + (3º Quartil - 1º Quartil) * 1'5) + i el mínim = (1º Quartil - (3º Quartil - 1º Quartil) * 1'5). La distribució de dades dins la capsa es coneix com *ICR* (3º Quartil - 1º Quartil).
- Les dades més enllà del valors indicats pels "whiskers" seràn considerats com a possibles *"outliers"*.

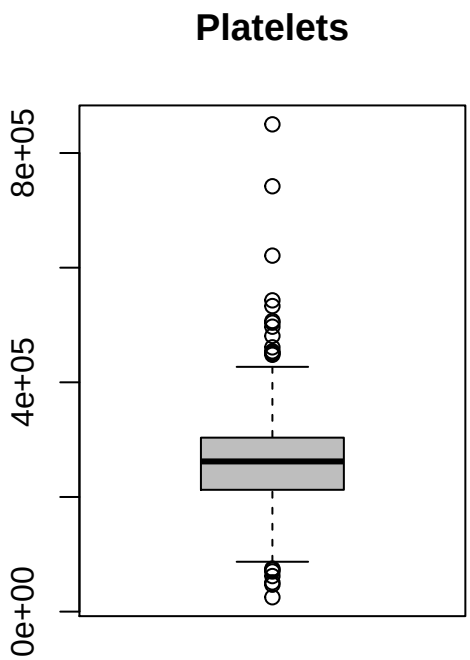
```
#Estudiem els outliers
attach(heart_data)
par(mfrow=c(1,2))
boxplot(age,main="Age", col="gray")
boxplot(creatinine_phosphokinase,main="Creatinine_phosphokinase", col="gray")
```



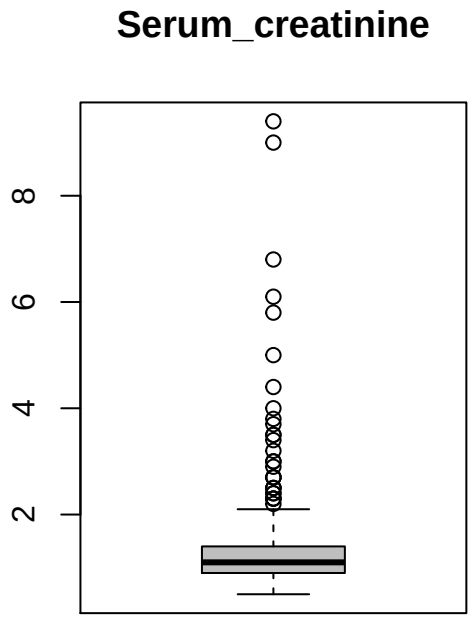


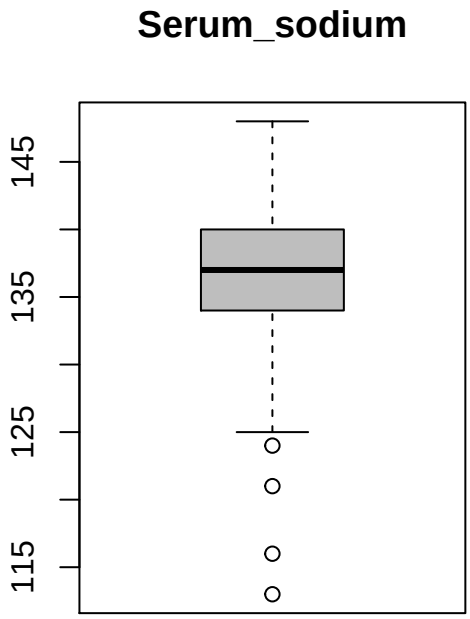

```
boxplot(ejection_fraction,main="Ejection_fraction", col="gray")
boxplot(platelets,main="Platelets", col="gray")
```



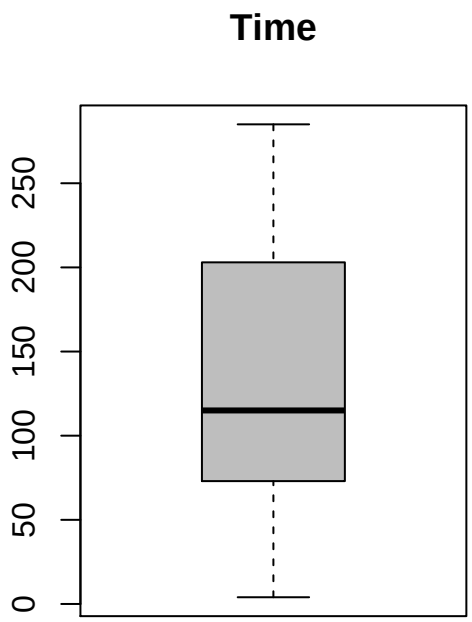



```
boxplot(serum_creatinine,main="Serum_creatinine", col="gray")
boxplot(serum_sodium,main="Serum_sodium", col="gray")
```






```
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
```

```
#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
```

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

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

```
#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.

4 Anàlisi de les dades.

4.1 Selecció dels grups de dades que es volen analitzar/comparar

Ara passarem a definir alguns subgrups de dades del nostre dataset que creiem que poden ser interessants a l'hora de l'anàlisi. L'elecció la farem tenim en ment algunes preguntes que ens volem respondre, com per exemple:

- Tenen més edat els pacients que acaben morint.
- Tenen un nivell major de `platelets` els pacients que acaben morint
- Tenen un major seguiment en dies els pacients que acaben morint.
- Tenen un nivell major de `creatinine_phosphokinase` els pacients que acaben morint.

Aprofitarem el mètode *"describeBy()"* que ens aportarà informació rellevant per conèixer per avançat dades referent a com es distribueixen les observacions dels atributs numèrics versus a quina classe pertanyen.

```
#Agrupació indicant si els pacients acaben morint o no
heart_data.SI.Death <- subset(heart_data, DEATH_EVENT == "Si")
heart_data.NO.Death <- subset(heart_data, DEATH_EVENT == "No")

#Agrupació per malatia
heart_data.diabetic <- subset(heart_data, diabetes == 'Si')
print(paste("Rows [Diabetes]: ", dim(heart_data.diabetic)[1]))
```

```
## [1] "Rows [Diabetes]:  125"
```

```
heart_data.hipertens <- subset(heart_data, high_blood_pressure == 'Si')
print(paste("Rows [High_Blood_Pressure]: ", dim(heart_data.hipertens)[1]))
```

```
## [1] "Rows [High_Blood_Pressure]:  105"
```

```
heart_data.anemic <- subset(heart_data, anaemia == 'Si')
print(paste("Rows [Anaemia]: ", dim(heart_data.anemic)[1]))
```

```
## [1] "Rows [Anaemia]:  129"
```

```
#Agrupacio per sexe
heart_data.home <- subset(heart_data, sex == 'masculi')
print(paste("Rows [Masculi]: ", dim(heart_data.home)[1]))
```

```
## [1] "Rows [Masculi]:  194"
```

```
heart_data.femeni <- subset(heart_data, sex == 'femeni')
print(paste("Rows [Femeni]: ", dim(heart_data.femeni)[1]))
```

```
## [1] "Rows [Femeni]:  105"
```

```
#Agrupació per fumador
heart_data.fumador <- subset(heart_data, smoking == 'Si')
print(paste("Rows [Smoking]: ", dim(heart_data.fumador)[1]))
```

```
## [1] "Rows [Smoking]:  96"
```

```
#Agrupacio per mort
heart_data.mort <- subset(heart_data, DEATH_EVENT == 'Si')
print(paste("Rows [Death]: ", dim(heart_data.mort)[1]))
```

```
## [1] "Rows [Death]:  96"
```

Obtenim informació de les dimensions numèriques en funció de a quina classe pertanyi cadascuna de les observacions que formen el vector, podem veure dades com:

- Mean, Median, sd, min, max, number of elements.

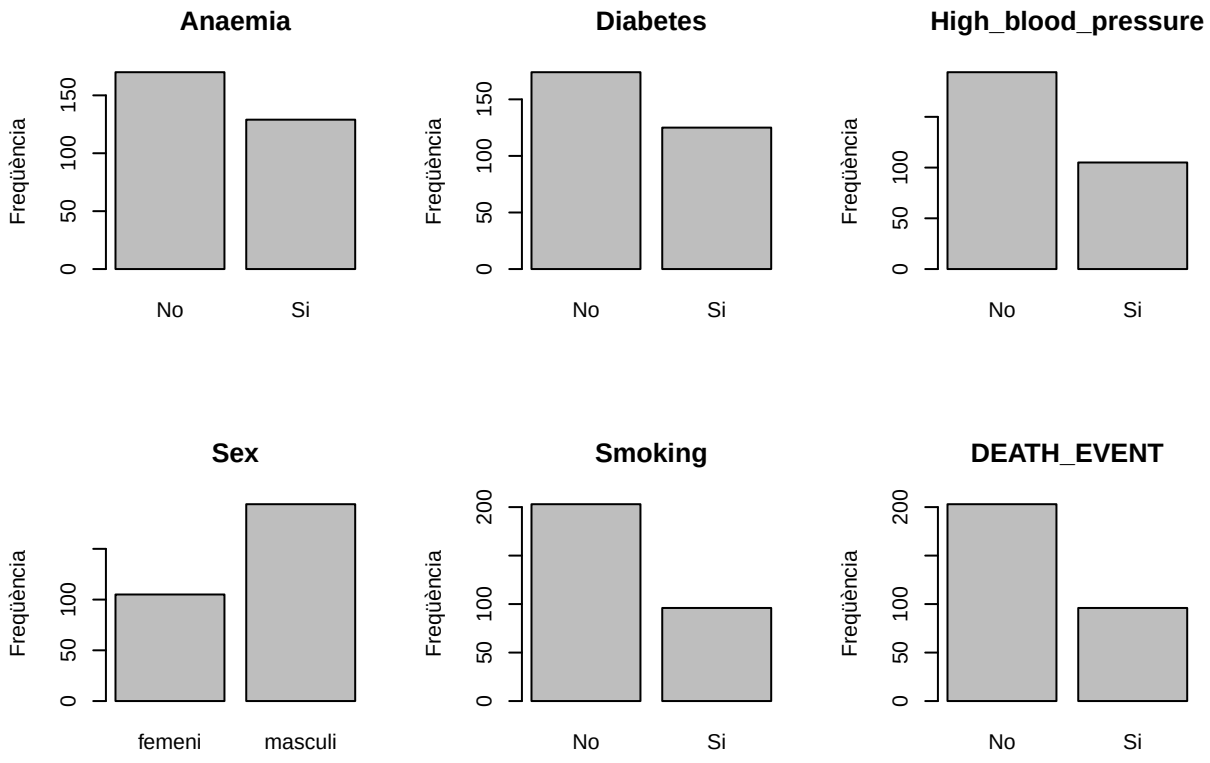
El següent pas serà comprovar si les dades segueixen una distribució normal i l'homogeneïtat de les variàncies. Per a fer-ho observarem les distribucions de forma visual i després corroborarem el que observem mitjançant testos estadístics.

4.2 Comprovació de la normalitat i homogeneïtat de la variància.

4.2.1 Variables categòriques

Visualitzarem i analitzarem les distribucions de les variables categòriques:

```
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.

4.2.2 Variables numèriques

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

Hem de tenir present que el nivell de confiança escollit és del 95%, això s'ha de tenir present per a les assumpcions de normalitat i homogeneïtat de la variança.

Per a analitzar si aquestes variables segueixen una distribució normal, primer observarem amb un gràfic la seva distribució, a continuació observarem també visualment si les dades segueixen una tendència normal i finalment realitzarem un test per a verificar definitivament si la variable segueix una distribució normal o no. El test que haurem d'aplicar és el test de *Lilliefors* perquè desconeixem la mitjana poblacional i tenim més de 50 observacions de la variable a estudiar.

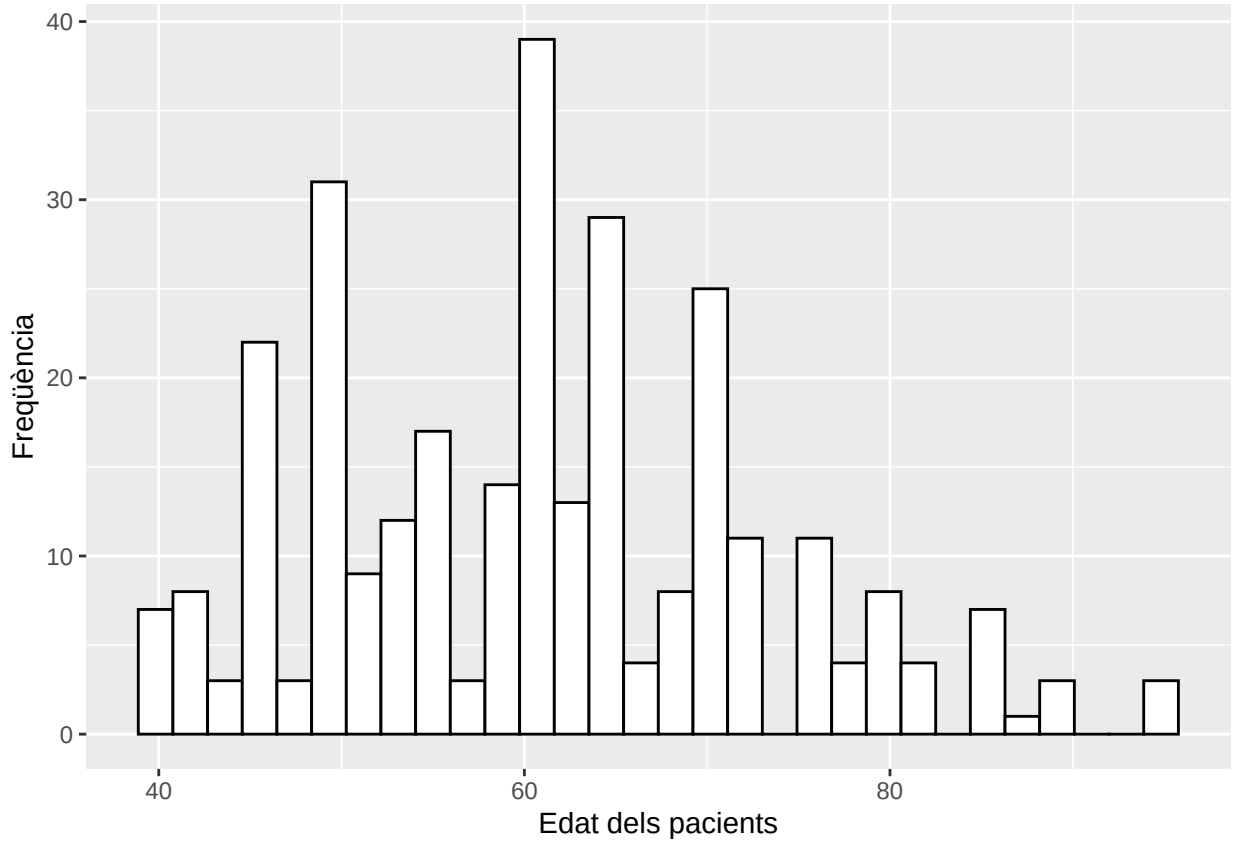
Per comprovar la homogeneïtat de la variança, utilitzarem el test de *Fligner-Killen*, aquest treballa amb 2 grups i per tant s'adapta perfectament a les nostres dades degut a que l'atribut que denota a quina classe pertanyen les diferents mostres només té 2 grups. Aquest test assumeix **l'igualtat de variances** entre els diferents grups de dades a l'hipòtesi nul · la (H_0) i per tant valors de probabilitat per sota el nivell de confiança ($< 0'05$) indicaran refusar la H_0 i per tant assumir que existeix heteroscedasticitat.

4.2.2.1 age

a) Normalitat

```
library(ggplot2)
ggplot(heart_data, aes(x = age)) + geom_histogram(fill="white",colour="black") +
  ylab("Freqüència") + xlab("Edat dels pacients")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

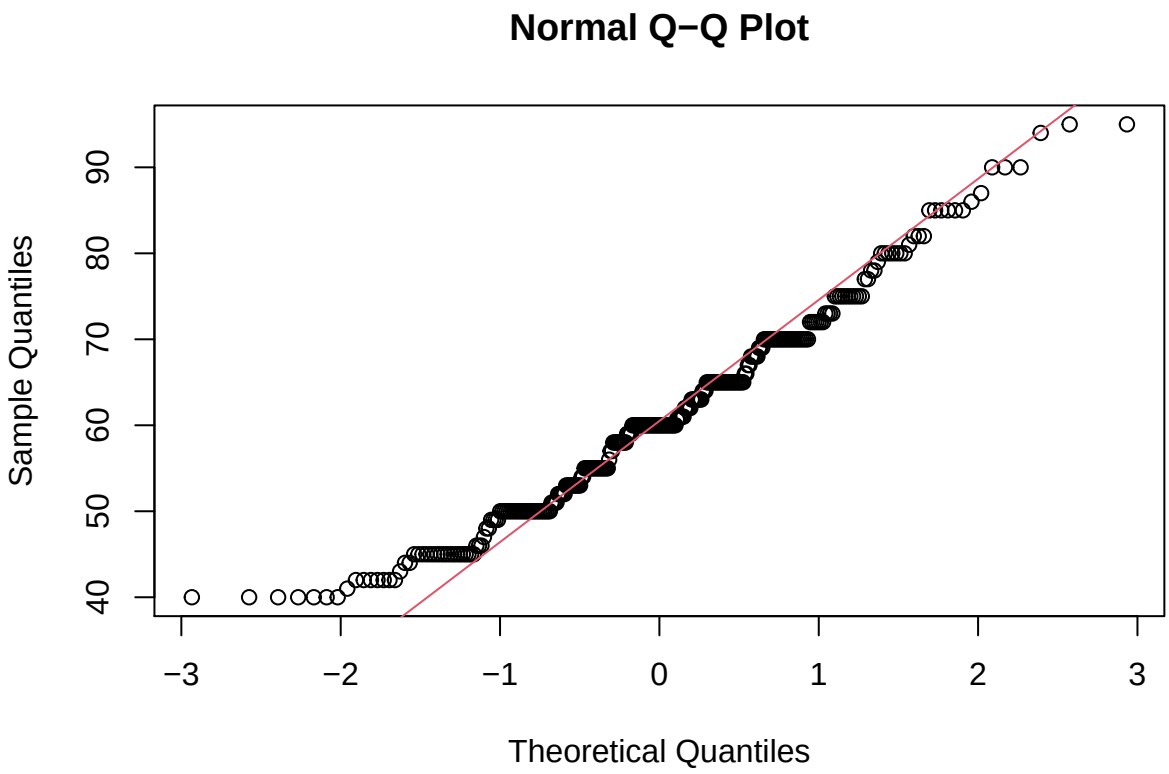


Veiem que la distribució, no sembla que segueixi una distribució normal perquè tenim molts alts i baixos, tot i que sí que té forma de campana de Gauss. Definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat. Tant per aquesta variable, com les següents, assumirem 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)
```



Per avaluar si les observacions de l'atribut **age** segueixen una distribució Normal hem utilitzat la funció **qqnorm()**. Aquesta el que fa és ordenar les dades del nostre atribut contra els quantils de una teorica distribució Normal. Els quantils dibuixats al eix X són calculats considerant una distribució Normal amb una mitja = 0 i una desviació standard = 1.

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
```

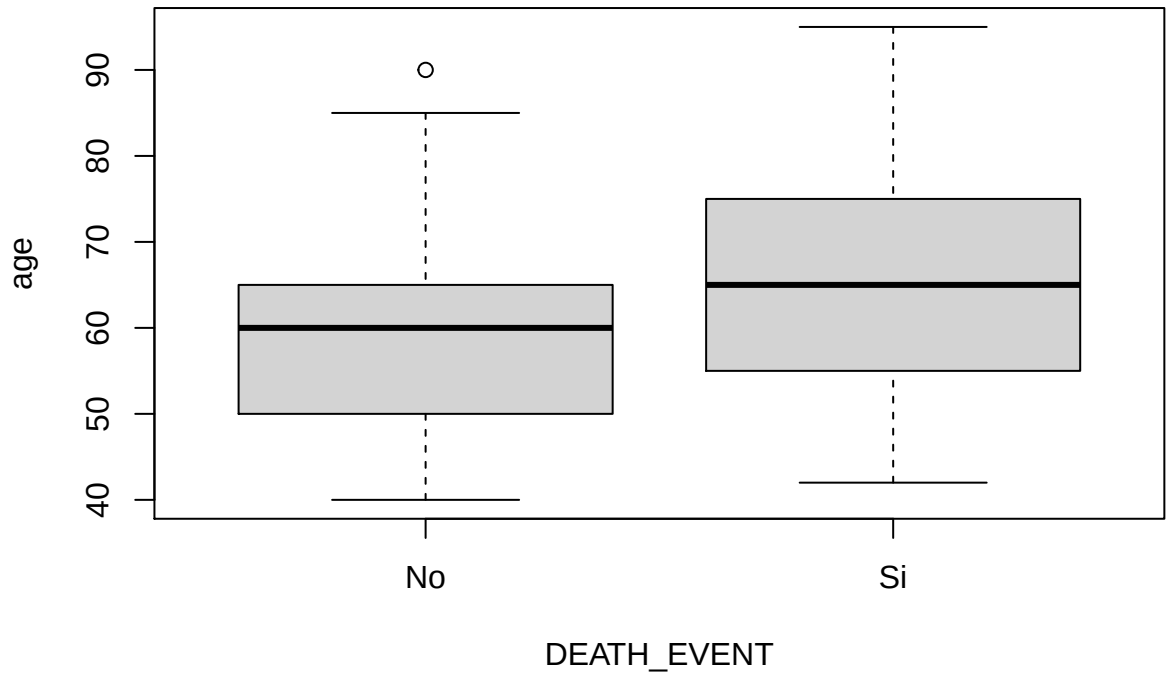
El test de Lilliefors ens dóna que el p-valor = 0.001304. Perquè segueixi normalitat amb un nivell de confiança del 95%, el p-valor ha de ser superior a 0.05. Com $0.001304 < 0.05$. Rebutgem la H_0 i diem que la variable *age* no segueix una distribució normal.

b) Homogeneïtat de la variança

```r
#Declarem llibreria necessària
if (!require(psych)) {install.packages("psych")}
library(psych)
#Informació de l'atribut age respecte a la classe a la qual pertany
describeBy(heart_data$age, heart_data$DEATH_EVENT)
```

```
##
##  Descriptive statistics by group
## group: No
##    vars   n  mean    sd median trimmed   mad min max range skew kurtosis   se
## X1    1 203 58.76 10.64     60   58.44 11.86  40  90    50 0.26    -0.39 0.75
## ------------------------------------------------------------
## group: Si
##    vars  n  mean    sd median trimmed   mad min max range skew kurtosis   se
## X1    1 96 65.22 13.21     65   64.71 14.83  42  95    53 0.29    -0.71 1.35
```

```r
plot (age ~ DEATH_EVENT, data = heart_data)
```




```r
fligner.test(age ~ DEATH_EVENT, data = heart_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  age by DEATH_EVENT
## Fligner-Killeen:med chi-squared = 7.2715, df = 1, p-value = 0.007006
```

Les observacions de la dimensió `age` que pertanyen a la classe "Si" de la dimensió `death_event` sembla que

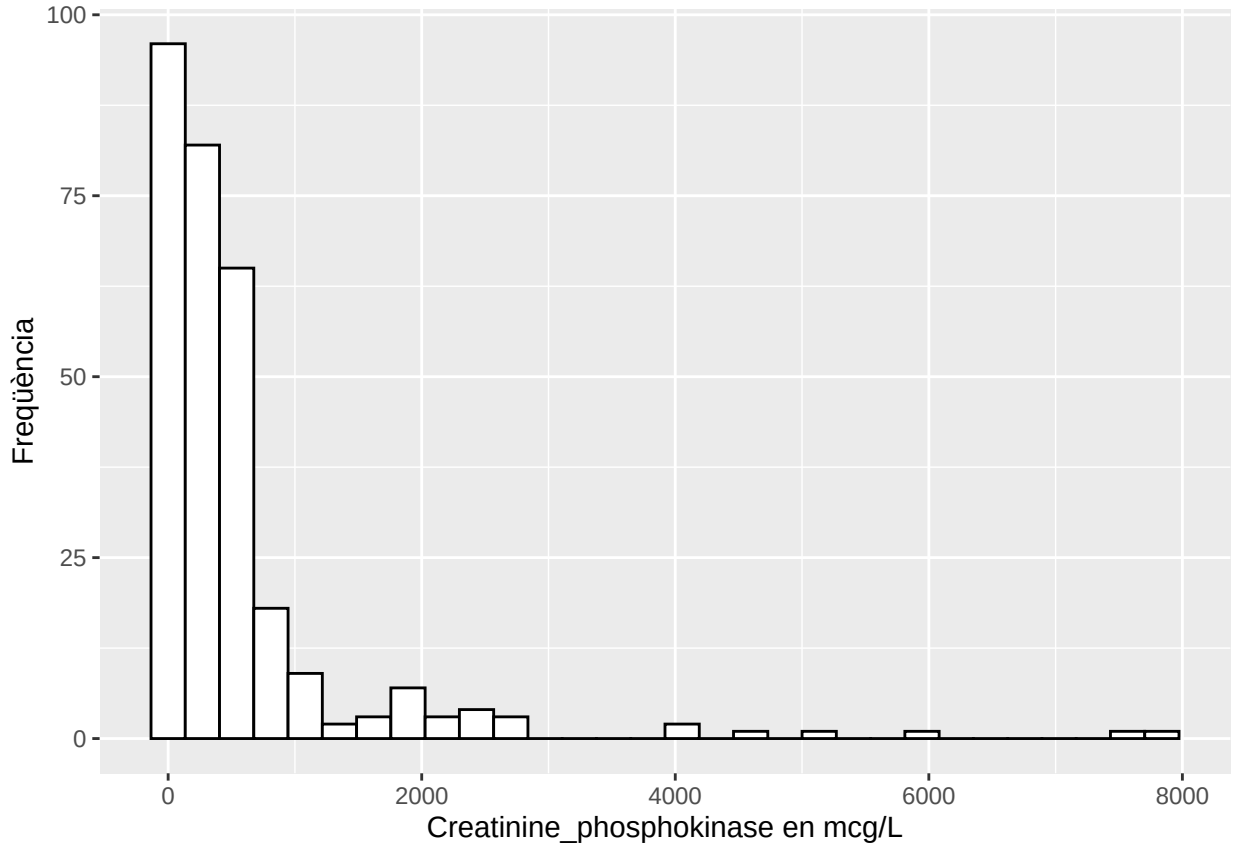
poden seguir una distribució Normal, mentre les altres no. Mirant amplada de la capsa i l'amplitud d'extrem a extrem podríem aproximar que pot haver una similitud de variances, però s'haurà de contrastar. Després d'aplicar el test observem que és compleix la condició $0.007006 < 0.05$ i per tant es refusa H_0 o sigui es compleix el principi de NO Homocedasticitat.

4.2.2.2 creatinine_phosphokinase

a) Normalitat

```
ggplot(heart_data, aes(x = creatinine_phosphokinase)) +
  geom_histogram(fill="white",colour="black") + ylab("Freqüència") +
  xlab("Creatinine_phosphokinase en mcg/L")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

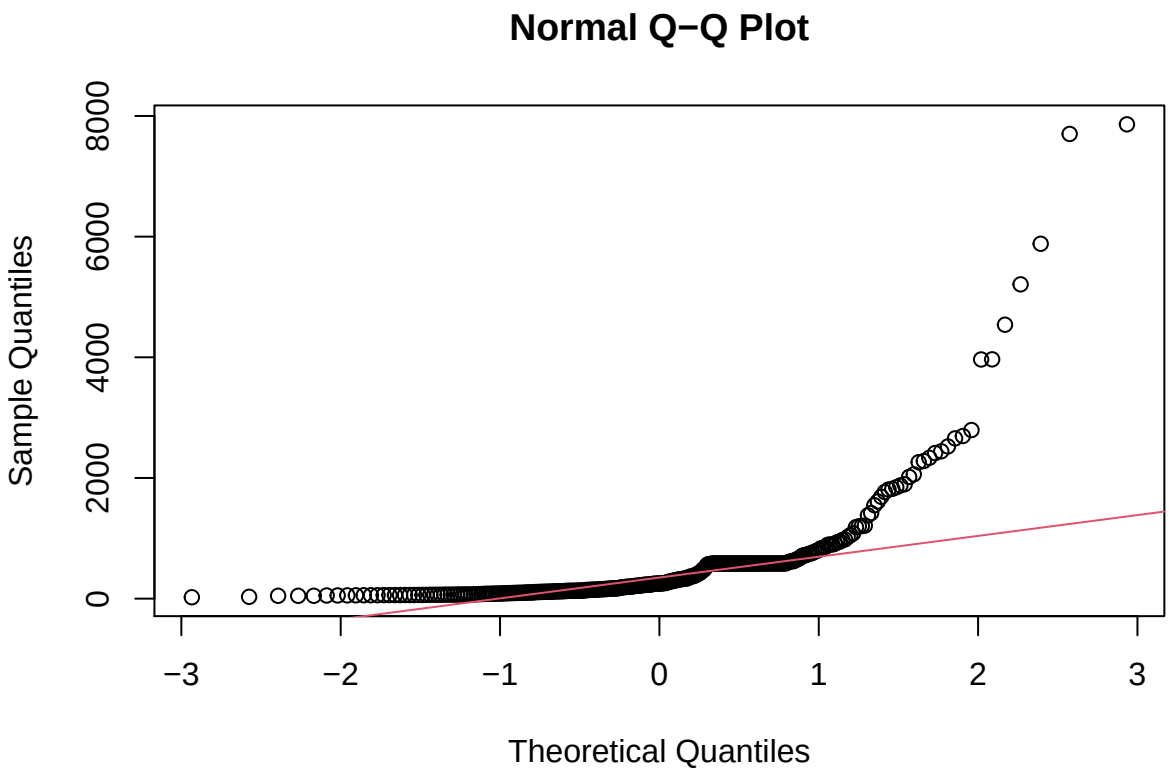


No sembla que segueixi una distribució normal. Definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

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.

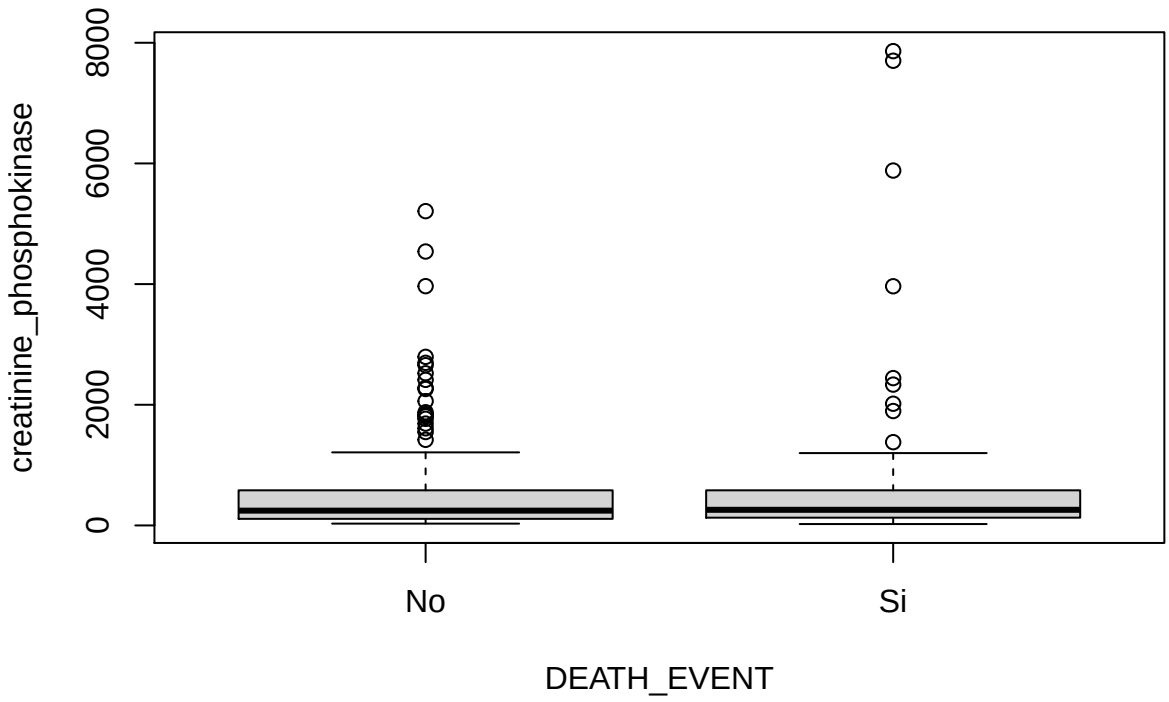
b) Homogeneïtat de la variança

```
#Informació de l'atribut [creatinine_phosphokinase] respecte a la classe a la qual pertany
describeBy(heart_data$creatinine_phosphokinase, heart_data$DEATH_EVENT)
```

```
##
##  Descriptive statistics by group
## group: No
##    vars   n   mean    sd median trimmed    mad min  max range skew kurtosis
## X1    1 203 540.05 753.8    245  366.39 268.35  30 5209  5179 3.15    12.41
##       se
## X1 52.91
```

```
## ------------------------------------------------------------
## group: Si
##    vars  n  mean      sd median trimmed    mad min  max range skew kurtosis
## X1    1 96 670.2 1316.58    259  363.47 256.49  23 7861  7838 4.13    17.81
##        se
## X1 134.37
```

```
plot (creatinine_phosphokinase ~ DEATH_EVENT, data = heart_data)
```




```
fligner.test(creatinine_phosphokinase ~ DEATH_EVENT, data = heart_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  creatinine_phosphokinase by DEATH_EVENT
## Fligner-Killeen:med chi-squared = 0.26845, df = 1, p-value = 0.6044
```

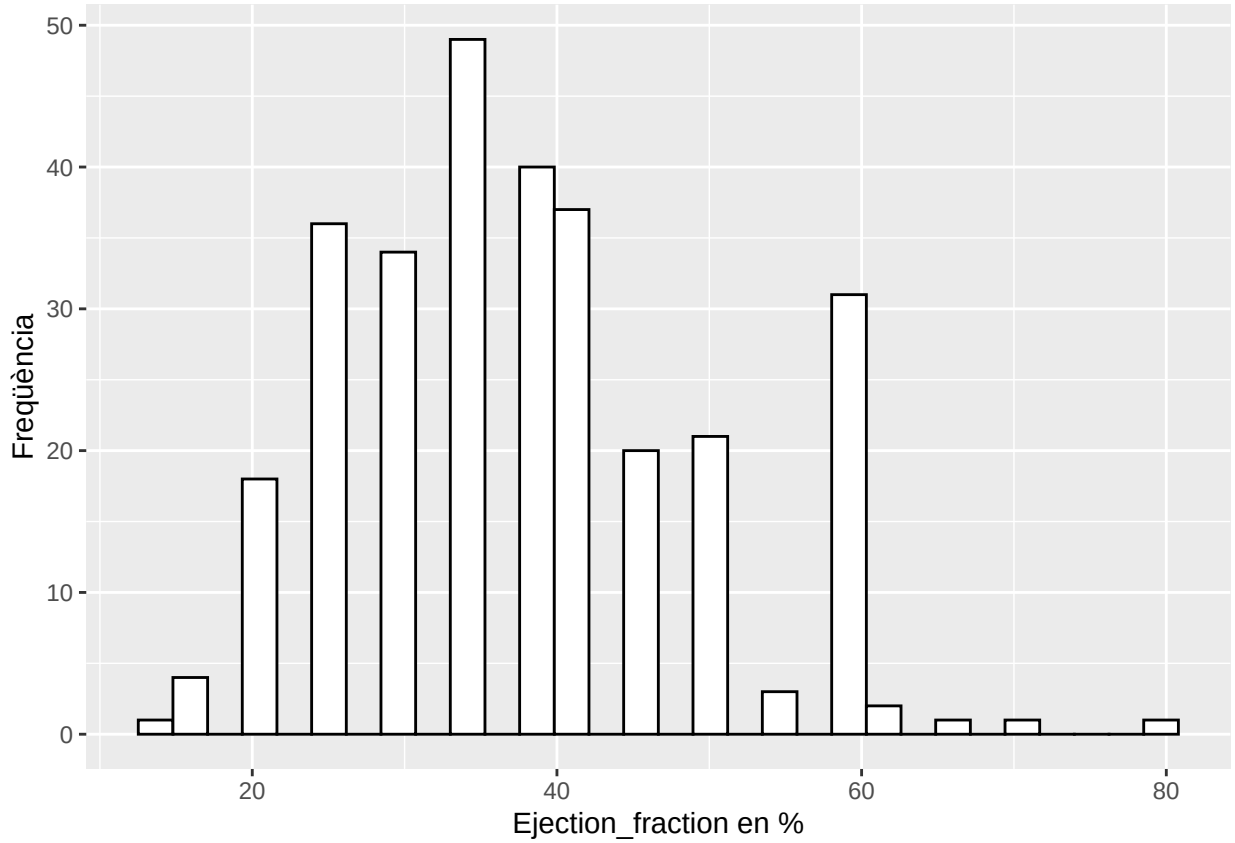
En la dimensió `creatinine_phosphokinase` pràcticament les 2 distribucions segueixen el mateix patró. Després d'aplicar el test test de *Fligner-Killen* observem que NO es compleix la condició $0.6044 < 0.05$ i per tant NO es rebutja la H_0, concloem que es compleix el principi d'homoscedasticitat.

4.2.2.3 ejection_fraction

a) Normalitat

```
ggplot(heart_data, aes(x = ejection_fraction)) +
  geom_histogram(fill="white",colour="black") +
  ylab("Freqüència") + xlab("Ejection_fraction en %")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

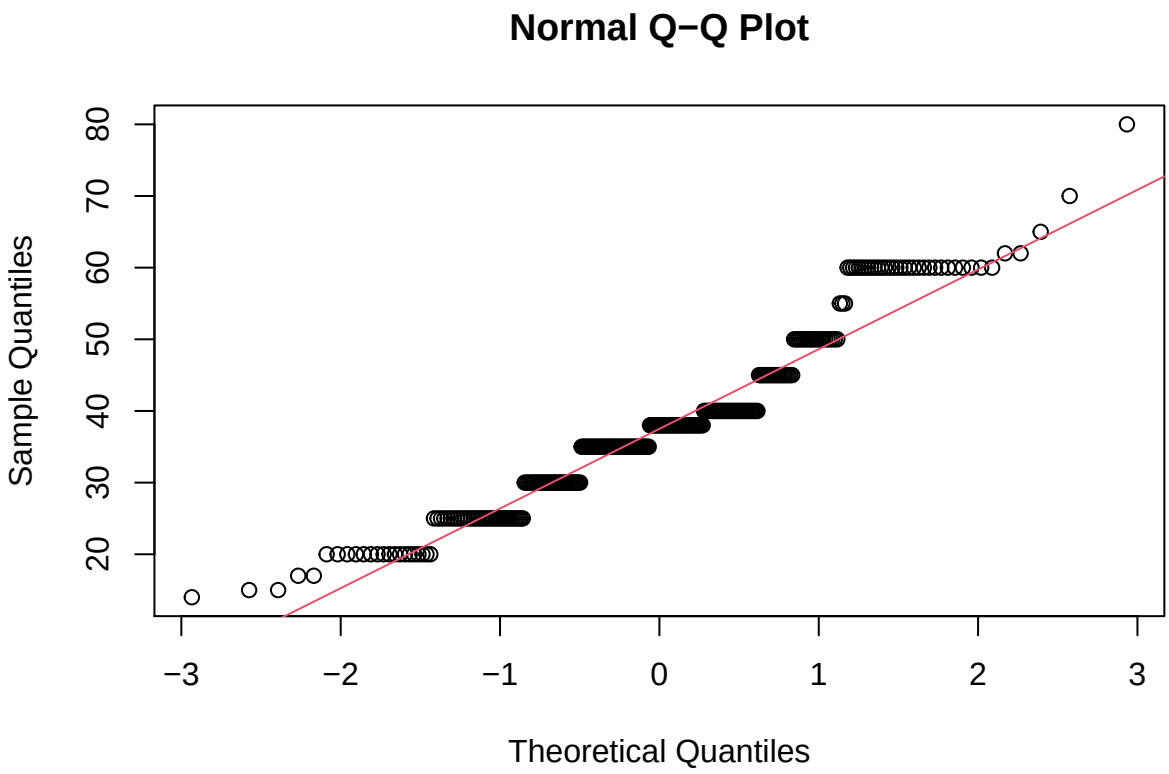


L'inici del gràfic si que sembla que segueix una distribució normal però a partir de la mediana sembla que no, per tant com en els apartats anteriors realitzarem un test per sortir de dubtes. Definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

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.

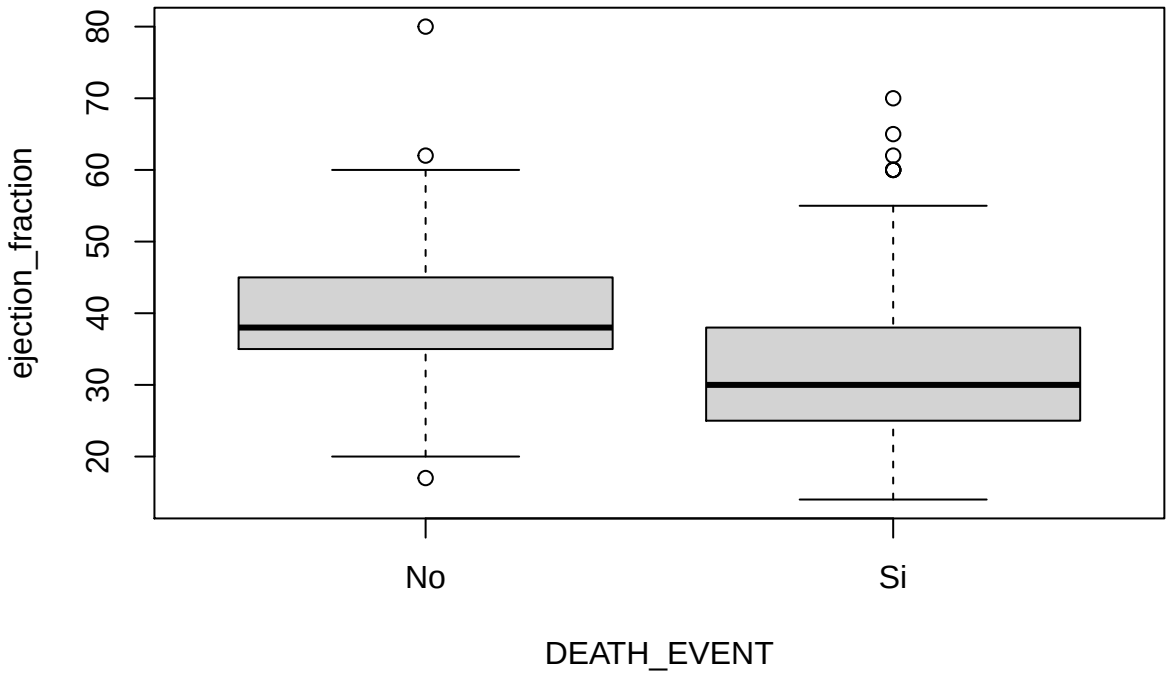
b) Homogeneïtat de la variança

```
#Informació de l'atribut ejection_fraction respecte a la classe a la qual pertany
describeBy(heart_data$ejection_fraction, heart_data$DEATH_EVENT)
```

```
##
##  Descriptive statistics by group
## group: No
##    vars   n  mean    sd median trimmed   mad min max range skew kurtosis   se
## X1    1 203 40.27 10.86     38   39.69 10.38  17  80    63 0.69     0.16 0.76
## ------------------------------------------------------------
## group: Si
```

```
##    vars  n  mean    sd median trimmed   mad min max range skew kurtosis   se
## X1    1 96 33.47 12.53     30   32.18 11.86  14  70    56  0.8     0.07 1.28
```

```r
plot (ejection_fraction ~ DEATH_EVENT, data = heart_data)
```




```r
fligner.test(ejection_fraction ~ DEATH_EVENT, data = heart_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  ejection_fraction by DEATH_EVENT
## Fligner-Killeen:med chi-squared = 4.5357, df = 1, p-value = 0.03319
```

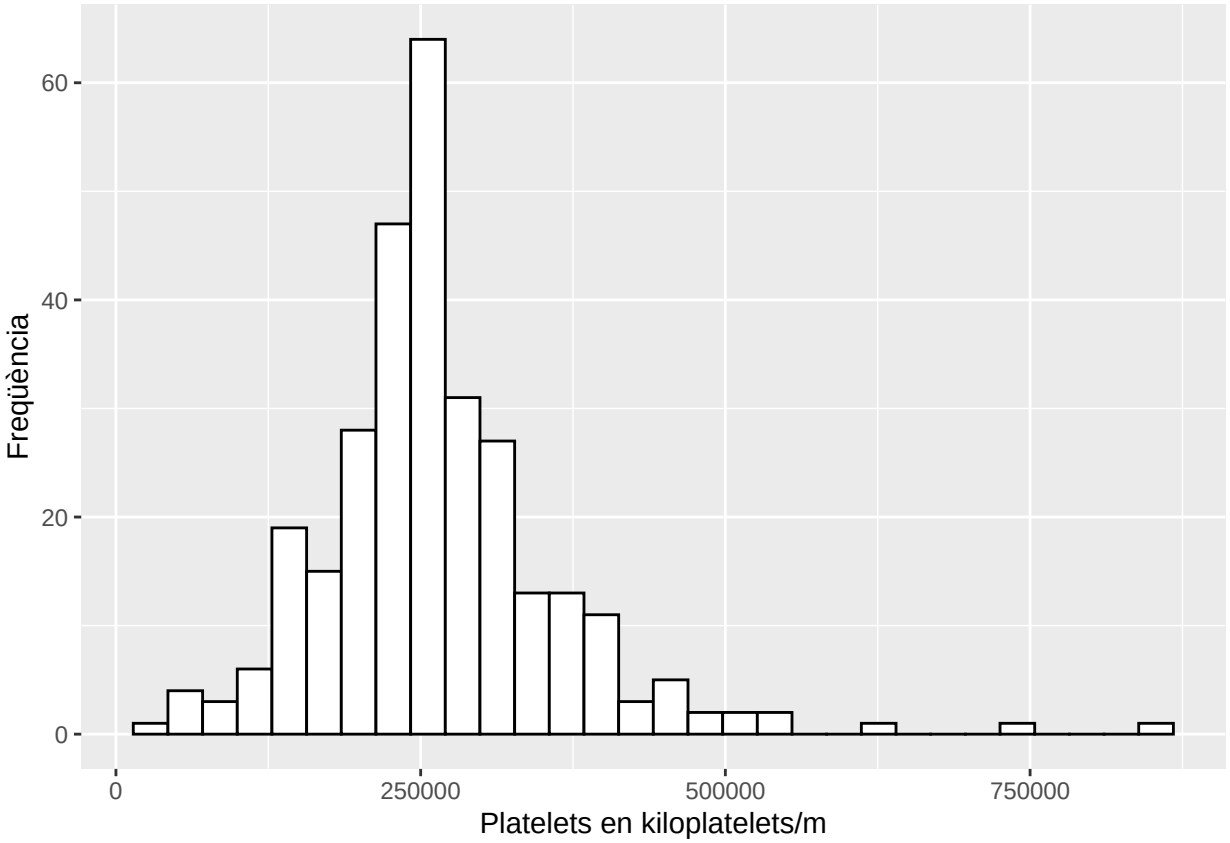
En la dimensió `ejection_fraction` la similitud de les variances semblen més dificils d'obtenir. Després d'aplicar el test observem que és compleix la condició $0.03319 < 0.05$ i per tant es refusa H_0, es compleix el principi de NO Homocedasticitat.

4.2.2.4 platelets

a) Normalitat

```r
ggplot(heart_data, aes(x = platelets)) + geom_histogram(fill="white",colour="black") +
  ylab("Freqüència") + xlab("Platelets en kiloplatelets/m")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

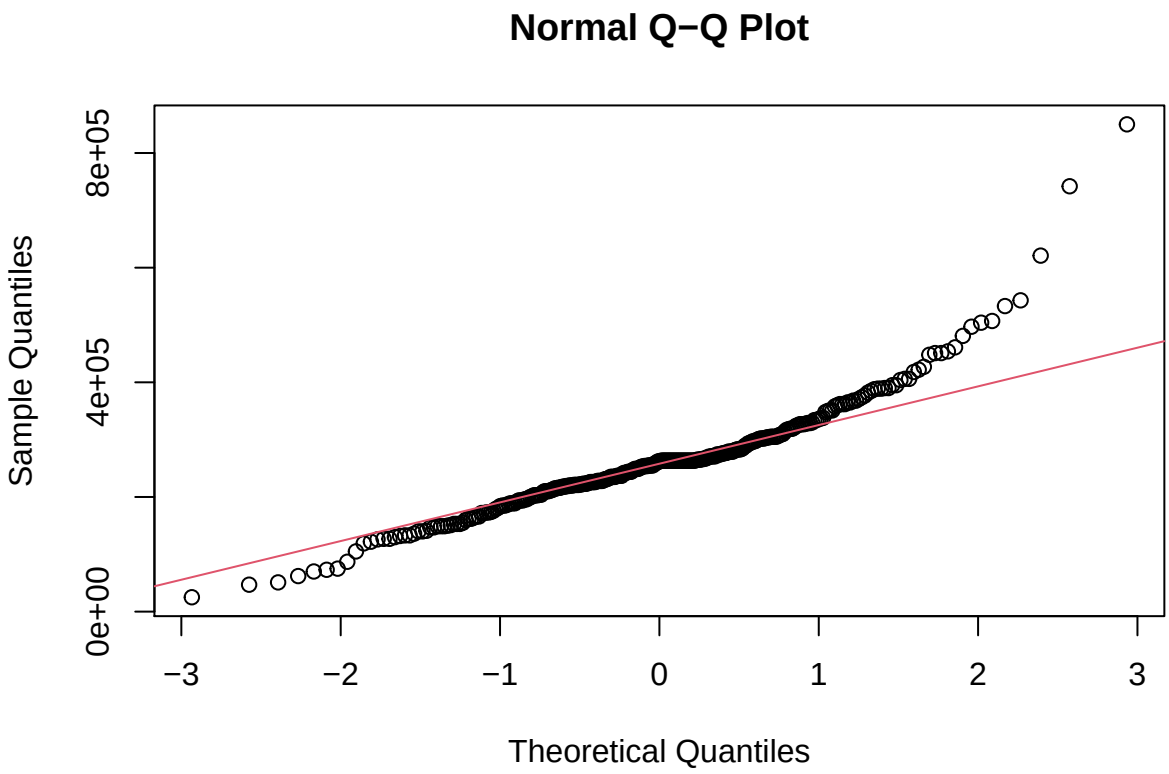


La distribució de la variable `platelets` succeeix igual que en l'apartat anterior, no sembla que segueixi una distribució normal. A continuació ho verificarem definint les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

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.

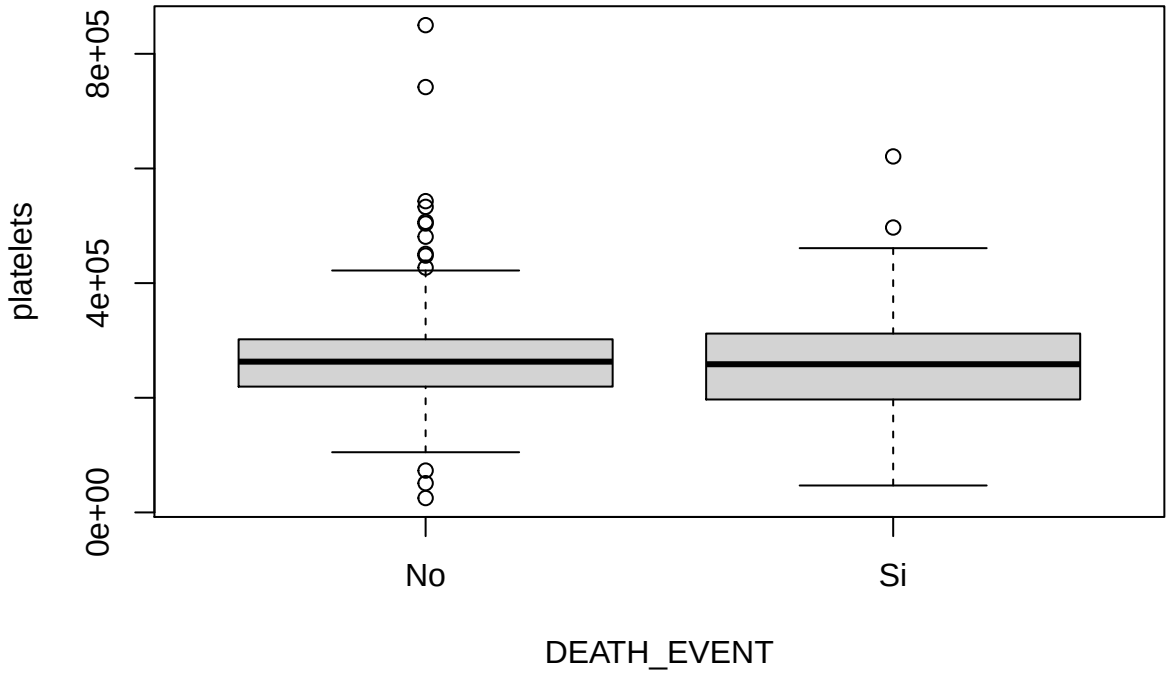
b) Homogeneïtat de la variança

```
#Informació de l'atribut platelets respecte a la classe a la qual pertany
describeBy(heart_data$platelets, heart_data$DEATH_EVENT)
```

```
##
##  Descriptive statistics by group
## group: No
##    vars   n     mean      sd median  trimmed     mad   min    max  range skew
## X1    1 203 266657.5 97531.2 263000 258707.8 62269.2 25100 850000 824900 1.86
##    kurtosis      se
## X1     8.21 6845.35
```

```
## ------------------------------------------------------------
## group: Si
##    vars  n   mean       sd median  trimmed     mad   min    max  range skew
## X1    1 96 256381 98525.68 258500 252404.9 88214.7 47000 621000 574000 0.61
##    kurtosis       se
## X1     1.15 10055.74
```

```
plot (platelets ~ DEATH_EVENT, data = heart_data)
```




```
fligner.test(platelets ~ DEATH_EVENT, data = heart_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  platelets by DEATH_EVENT
## Fligner-Killeen:med chi-squared = 2.3222, df = 1, p-value = 0.1275
```

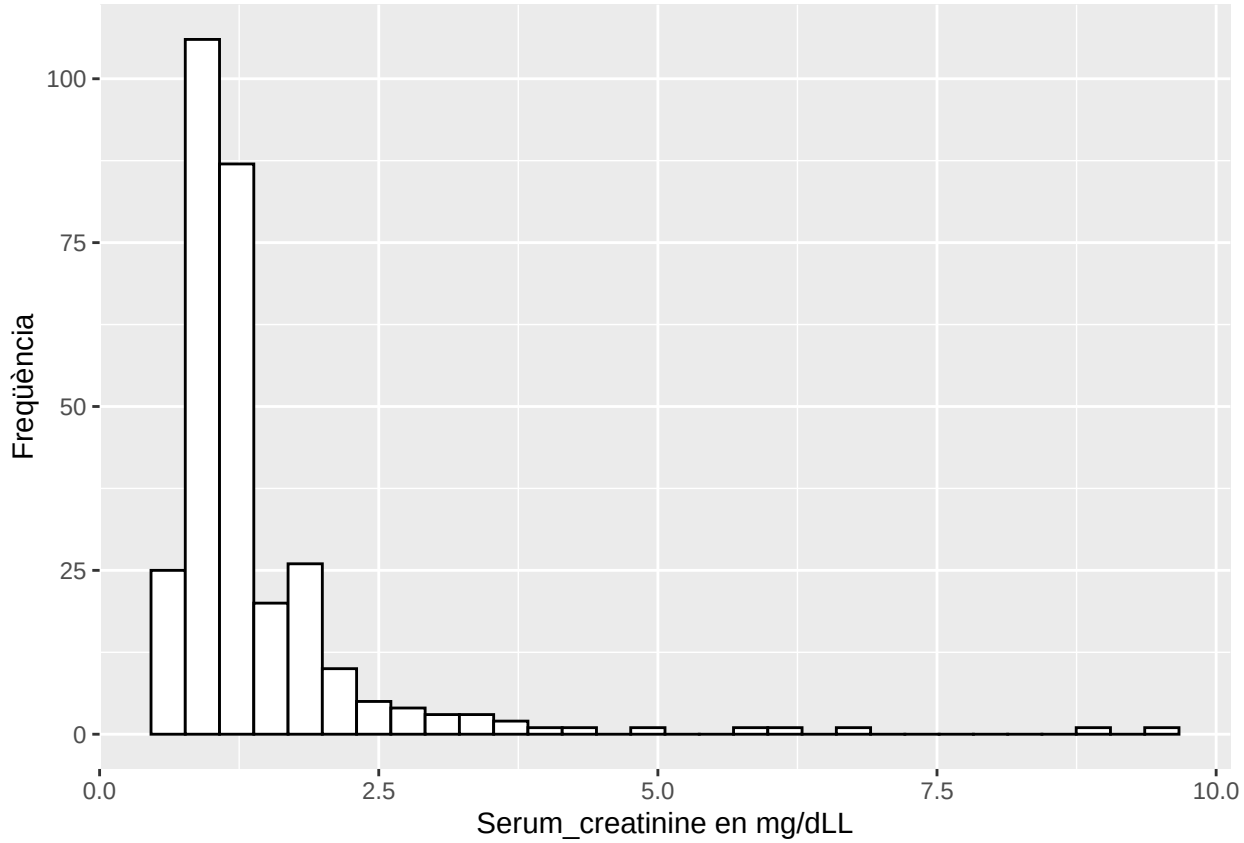
En la dimensió `platelets` les variances tenen certa similitud veient l'amplada de la capsa i la distància entre extrems. Després d'aplicar el test observem que NO és compleix la condició $0.1275 < 0.05$ i per tant NO es refusa H_0, és a dir, es compleix el principi de Homocedasticitat.

4.2.2.5 serum_creatinine

a) Normalitat

```
ggplot(heart_data, aes(x = serum_creatinine)) +
  geom_histogram(fill="white",colour="black") + ylab("Freqüència") +
  xlab("Serum_creatinine en mg/dLL")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

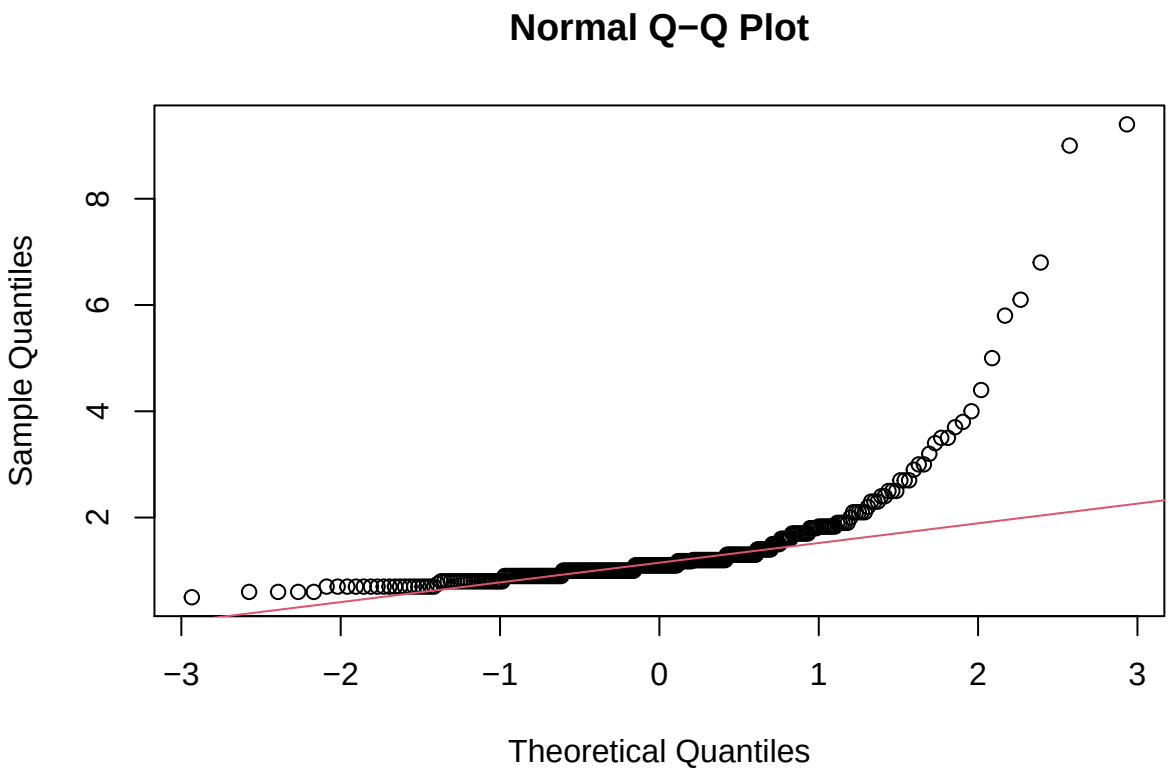


Observem clarament com la variable *serum_creatinine* no té forma de Campana de Gauss i no segueix una distribució normal. De totes maneres ens n'assegurarem definint les hipòtesis nul · la i alternativa i avaluant si compleix l'assumpció de normalitat.

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.

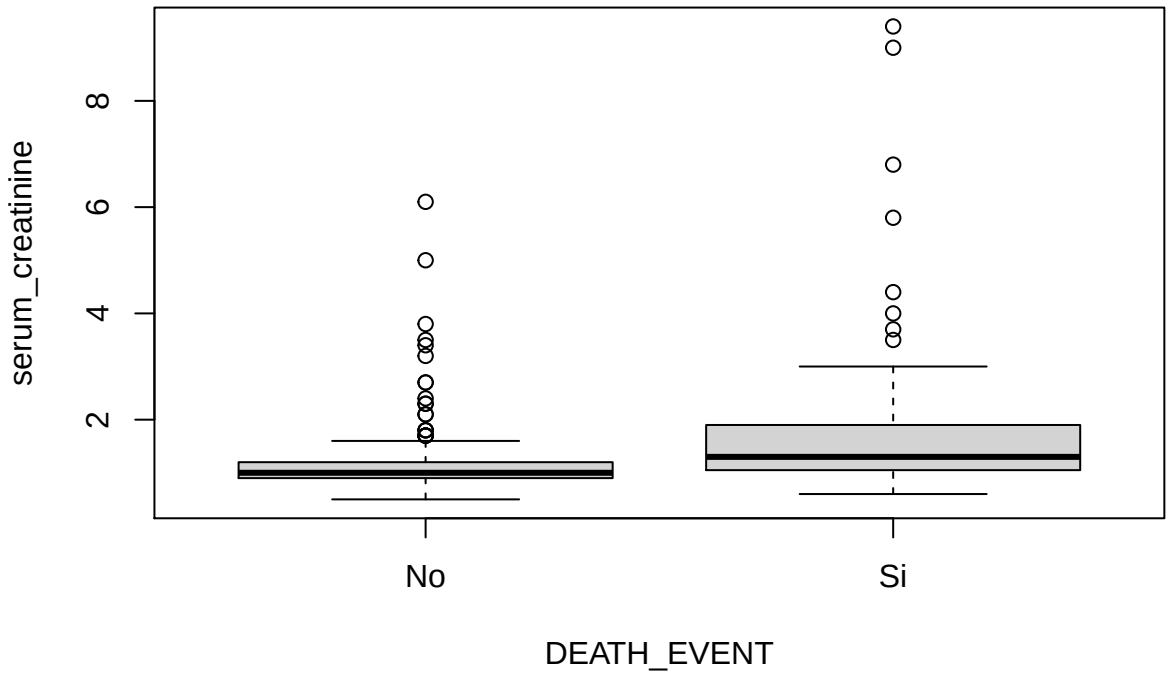
b) Homogeneïtat de la variança

```
#Informació de l'atribut serum_creatinine respecte a la classe a la qual pertany
describeBy(heart_data$serum_creatinine, heart_data$DEATH_EVENT)
```

```
##
##  Descriptive statistics by group
## group: No
##    vars   n mean   sd median trimmed mad min max range skew kurtosis   se
## X1    1 203 1.18 0.65      1    1.06 0.3 0.5 6.1   5.6 4.11    22.08 0.05
## ------------------------------------------------------------
## group: Si
```

```
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 96 1.84 1.47    1.3    1.53 0.59 0.6 9.4   8.8  3.3     12.4 0.15
```

```
plot (serum_creatinine ~ DEATH_EVENT, data = heart_data)
```




```
fligner.test(serum_creatinine ~ DEATH_EVENT, data = heart_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  serum_creatinine by DEATH_EVENT
## Fligner-Killeen:med chi-squared = 35.423, df = 1, p-value = 2.654e-09
```

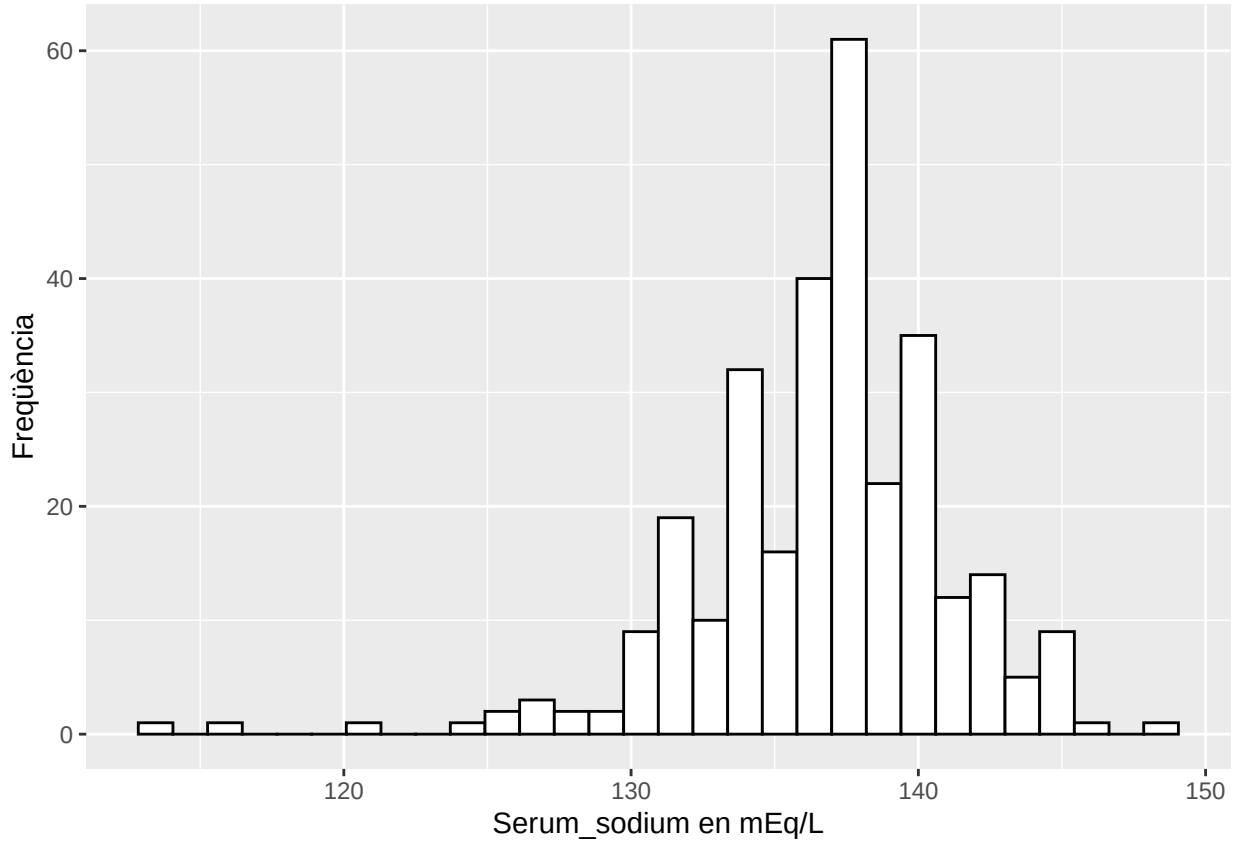
En la dimensió `serum_creatinine` s'observa poca similitud entre les variances. Després d'aplicar el test observem que és compleix la condició (2.654 * eˆ-9) < 0.05 i per tant es refusa H_0, és a dir, es compleix el principi de NO Homocedasticitat.

4.2.2.6 serum_sodium

a) Normalitat

```
ggplot(heart_data, aes(x = serum_sodium)) + geom_histogram(fill="white",colour="black") +
  ylab("Freqüència") + xlab("Serum_sodium en mEq/L")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

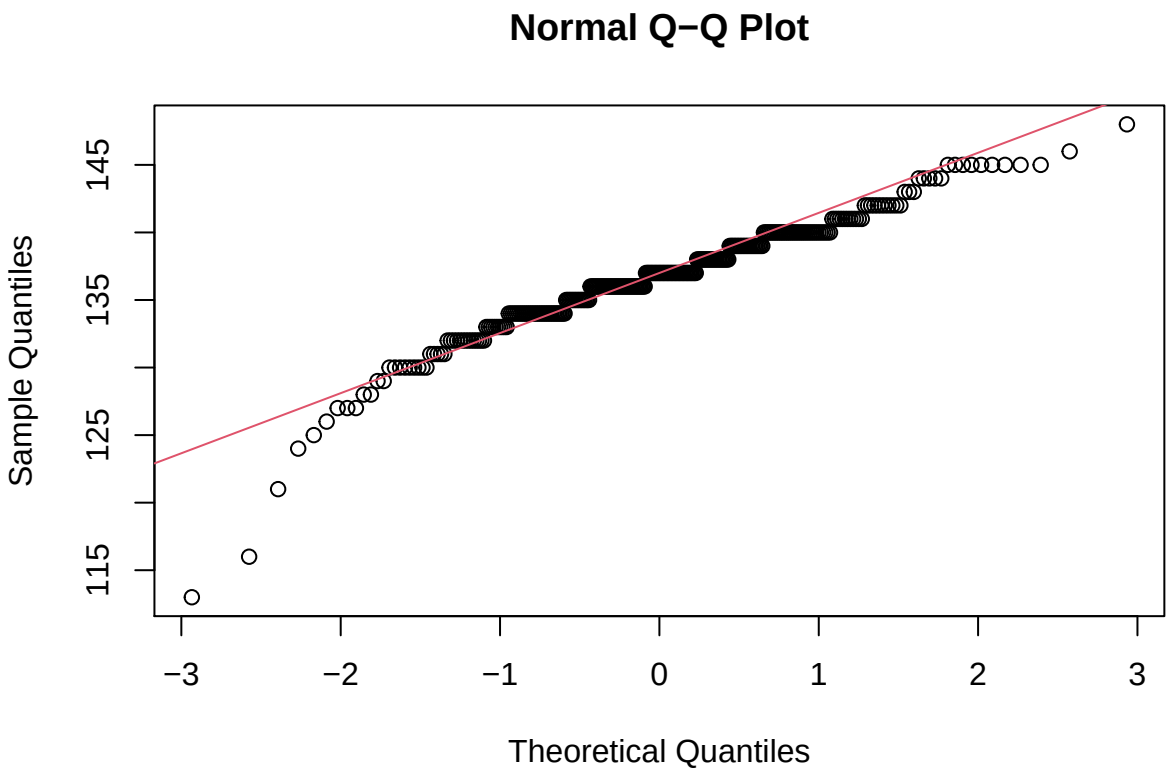


La distribució de la variable *serum_sodium* podria ser normal, per això ho comprovarem definint les hipòtesis nul · la i alternativa i avaluant si compleix l'assumpció de normalitat.

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.

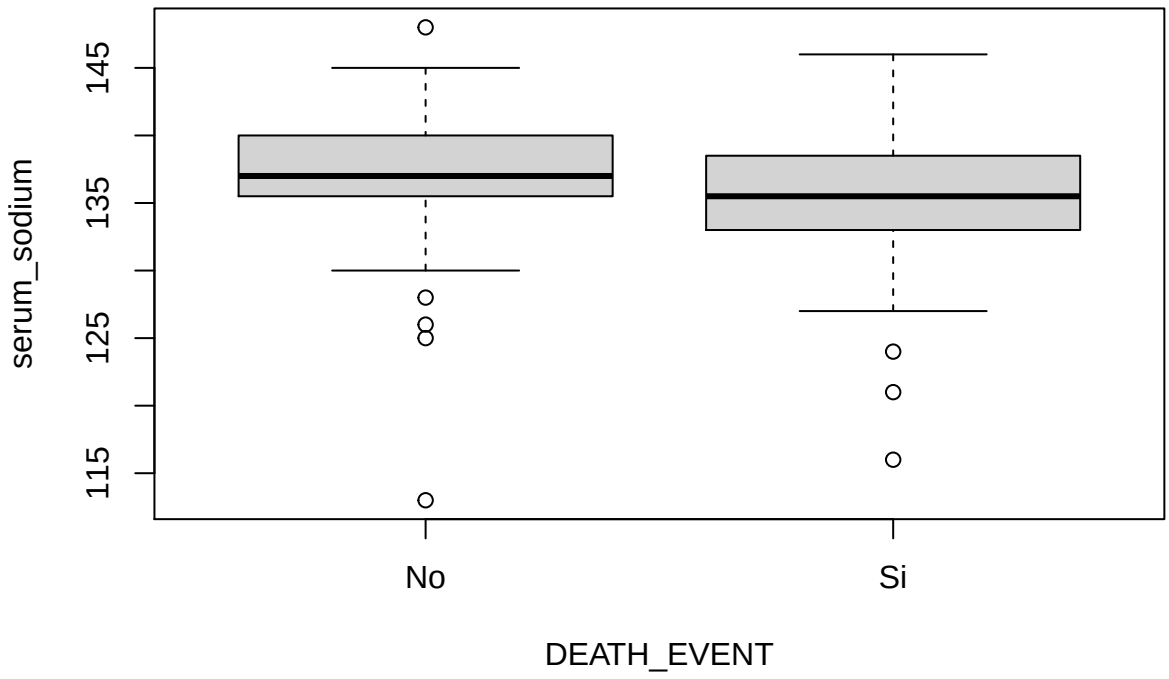
b) Homogeneïtat de la variança

```
#Informació de l'atribut serum_sodium respecte a la classe a la qual pertany
describeBy(heart_data$serum_sodium, heart_data$DEATH_EVENT)
```

```
##
##  Descriptive statistics by group
## group: No
##    vars   n   mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 203 137.22 3.98    137  137.36 2.97 113 148    35 -1.2     6.21 0.28
## ------------------------------------------------------------
```

```
## group: Si
##    vars  n   mean sd median trimmed  mad min max range  skew kurtosis   se
## X1    1 96 135.38  5  135.5  135.55 3.71 116 146    30 -0.66     1.81 0.51
```

```
plot (serum_sodium ~ DEATH_EVENT, data = heart_data)
```




```
fligner.test(serum_sodium ~ DEATH_EVENT, data = heart_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  serum_sodium by DEATH_EVENT
## Fligner-Killeen:med chi-squared = 6.1205, df = 1, p-value = 0.01336
```

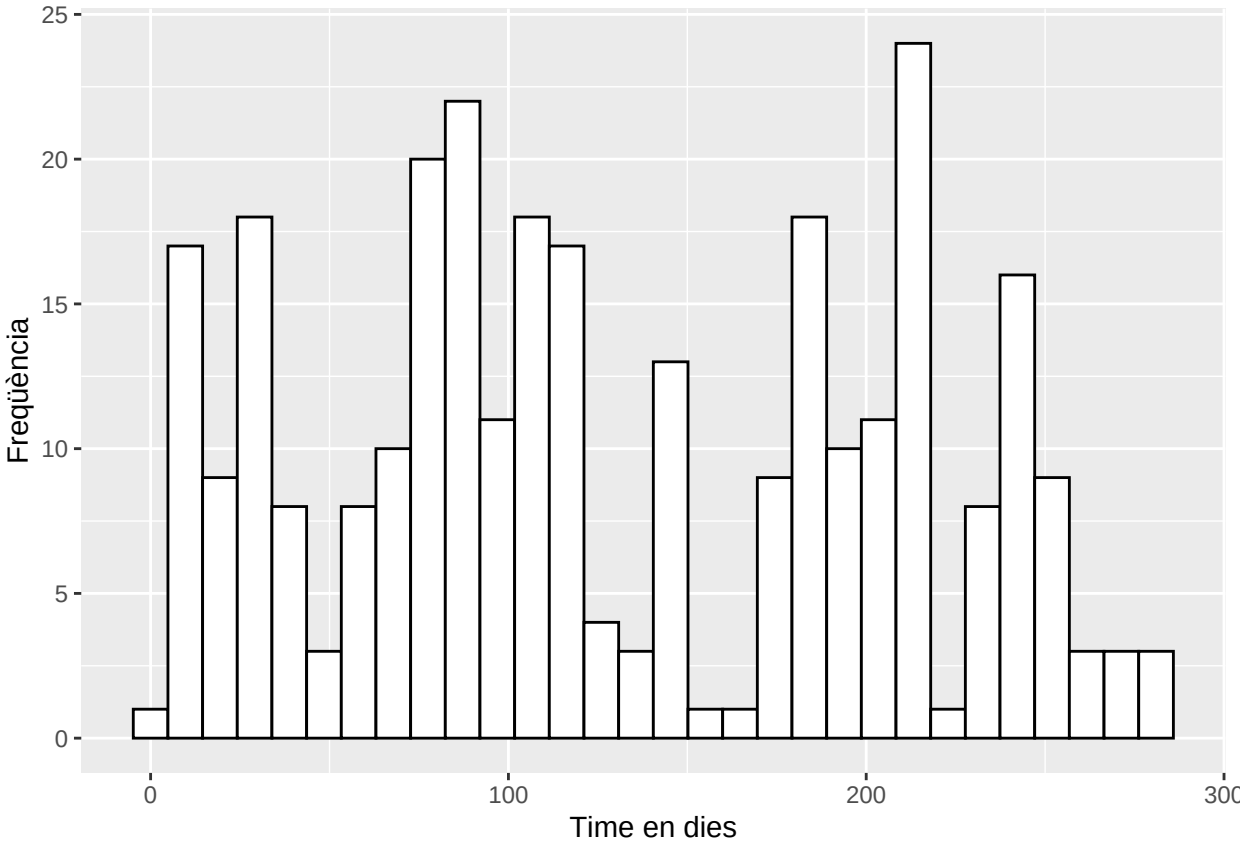
En la dimensió `serum_sodium` les observacions que pertanyen a la classe "Si" de la dimensió `death_event` l'amplada de capsa i distància entre extrems semblen iguals per les 2 distribucions, per tant podríen predir que les variances tenen certa similitud i quan avaluaem el test veiem que p-value = 0.01336 $<$ 0.05 i que per tant es refusa la H_0, per tant es compleix el principi de NO Homocedasticitat.

4.2.2.7 time

a) Normalitat

```
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") + xlab("Time en dies")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

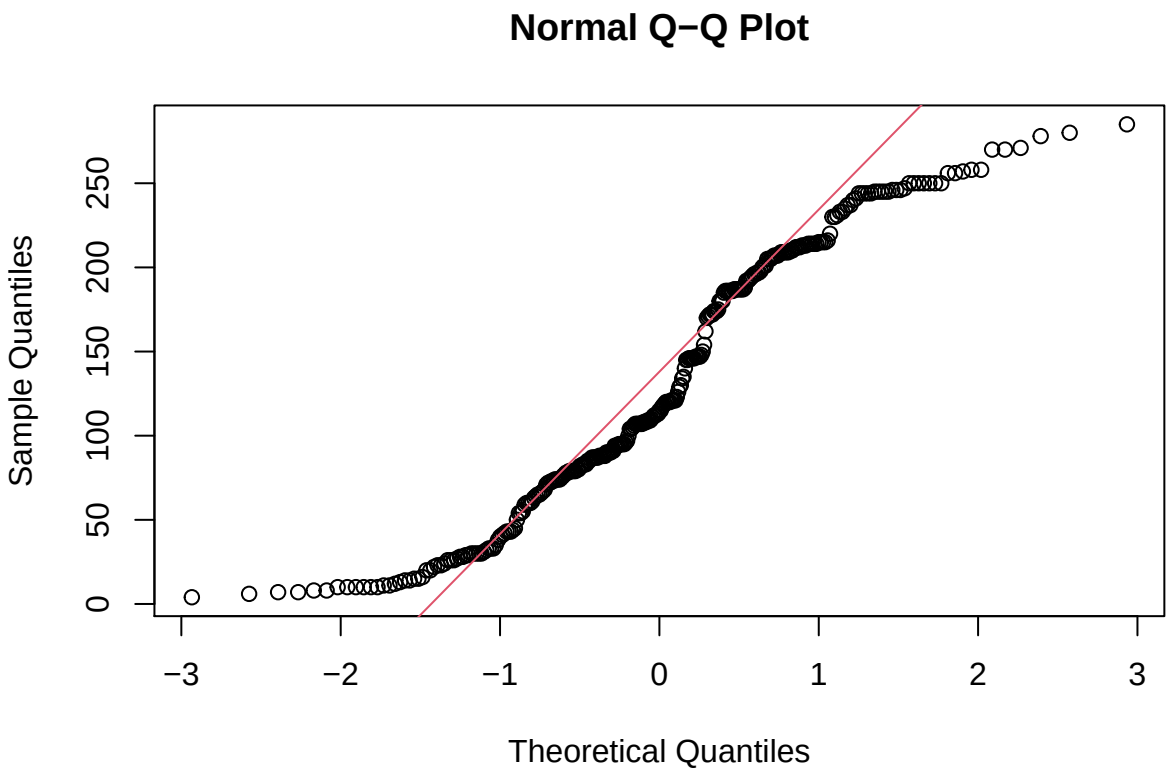


La distribució de la variable *time* no s'assembla en res a una distribució normal que té la forma d'una Campana de Gauss, però de totes maneres definirem les hipòtesis nul · la i alternativa i avaluarem si compleix l'assumpció de normalitat.

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)
```




```r
#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
```

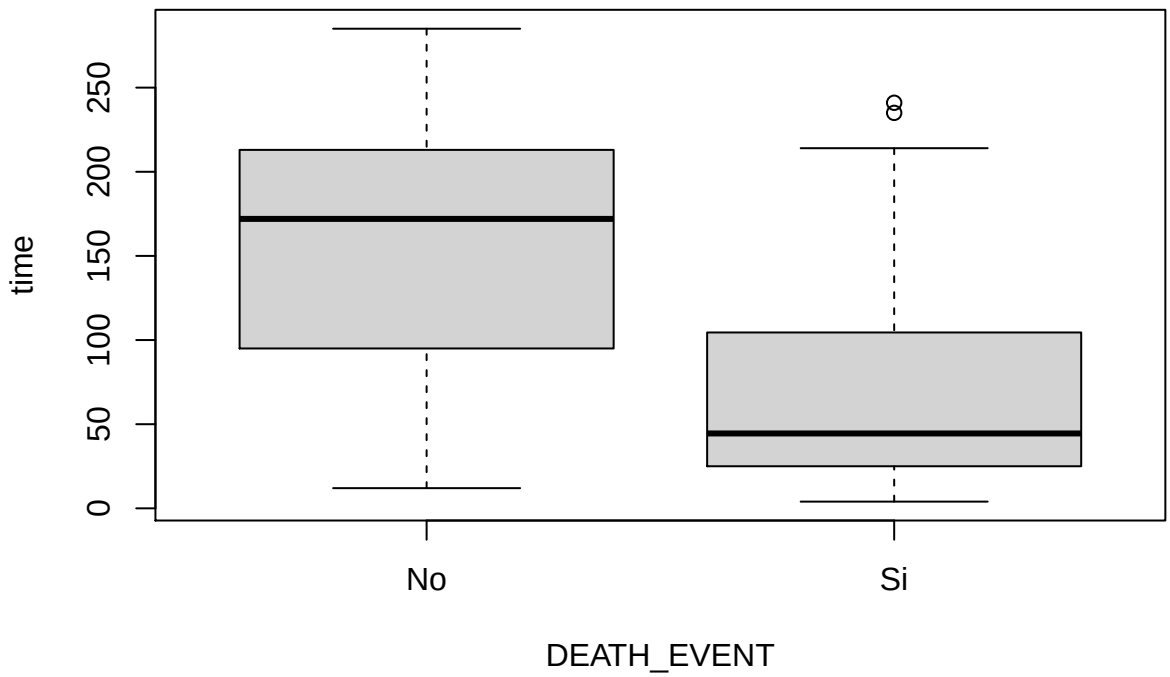
Efectivament en la representació de la *qqnorm*, les dades s'allunyen de la *qqline*. I el test de *Lilliefors* ens diu que el p-valor = 2.01e-08, per tant com aquest és més petit que 0.05, rebutgem la H_0 i diem que la variable *time* no segueix una distribució normal.

b) Homogeneïtat de la variança

```r
#Informació de l'atribut [time] respecte a la classe a la qual pertany
describeBy(heart_data$time, heart_data$DEATH_EVENT)
```

```
##
##  Descriptive statistics by group
## group: No
##    vars   n   mean    sd median trimmed   mad min max range  skew kurtosis   se
## X1    1 203 158.34 67.74    172   158.6 94.89  12 285   273 -0.05    -1.25 4.75
## ------------------------------------------------------------
## group: Si
##    vars  n  mean    sd median trimmed   mad min max range skew kurtosis   se
## X1    1 96 70.89 62.38   44.5   62.71 45.96   4 241   237 1.03    -0.04 6.37
```

```r
plot (time ~ DEATH_EVENT, data = heart_data)
```




```r
fligner.test(time ~ DEATH_EVENT, data = heart_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  time by DEATH_EVENT
## Fligner-Killeen:med chi-squared = 6.9936, df = 1, p-value = 0.00818
```

En l'anàlisis visual de la variable `time`, no s'observa similitud de les variances. Després d'aplicar el test observem que és compleix la condició $0.00818 < 0.05$ i per tant es refusa H_0, és a dir, es compleix el principi de NO Homocedasticitat.

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

4.3.1 Correlacions

Per veure les correlacions entre les diferents dimensions farem ús d'un *"scatter plot matrix"*, el qual ens proporcionarà una visualització de cada dimensió amb la resta.

A la part superior veurem l'scatter plot de les 2 variables avaluades i a la part inferior el valor del seu coeficient de correlació. Per obtenir aquest valor podíem optar entre el coeficient de correlació de Pearson i el coeficient de correlació de Spearman. Però l'elecció està condicionada als resultats de les proves realitzades prèviament i com ja hem vist cap de les variables numèriques segueix una distribució Normal i tampoc quasi cap d'elles compleix el principi d'homoscedasticitat i per tant haurem d'utilitzar el 2º coeficient, el coeficient de Spearman.

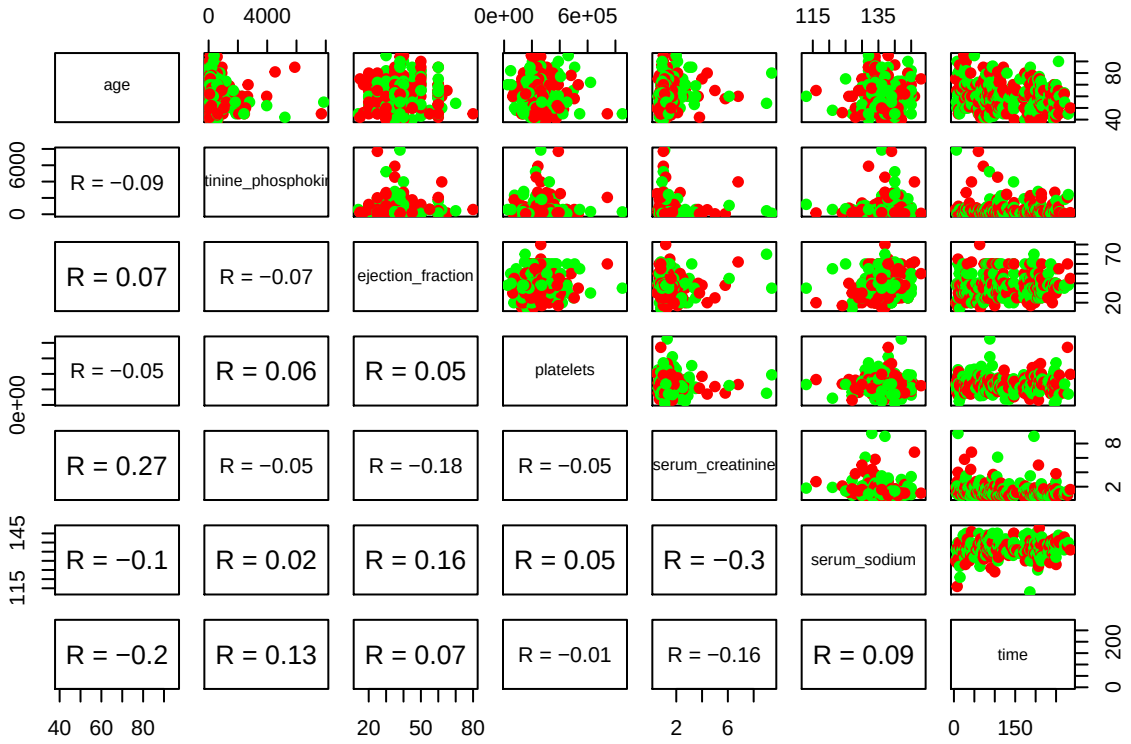
Veient les gràfiques superiors intuïm que pràcticament no existeix correlació entre cap parell de variables del nostre joc de dades, ho podem contrastar mirant a la part inferior i observant que quasi la majoria de valors obtinguts es mouen al voltant del valor 0 indicant l'absència de correlació. Si observem la correlació entre les variables i fem grups:

- Variables que tenen a veure amb la sang (`creatinine_phosphokinase`, `ejection_fraction`, `platelets`, `serum_creatinine`, `serum_sodium`).

- Variables que no tenen a veure amb la sang (`age` i `time`).

Veurem a priori que no existeix cap correlació entre `age` i `time` i la resta de variables, així com tampoc entre elles mateixes.

```
#Configuració del panel inferior
lower.panel <- function(x, y){
    par(usr = c(0, 1, 0, 1))        #Coord. de la regió de dibuix
    #Càlcul de la correlació
    r <- round(cor(x,y,use="complete.obs", method="spearman"), digits=2)
    txt <- paste0("R = ", r)        #txt -> R = valor correlació
    cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor)  #Dibuixem el valor de la correlació als gràfics
}
#Configuració del panel superior
#Indiquem l'aspecte dels punts que es dibuixaran en el panel
# pch = 19 -> solid circle
# col -> els colors que volem donar als punts
upper.panel<-function(x, y){
  points(x,y, pch = 19, col=c("red", "green"))
}
#Dibuixem els plots per visualitzar la correlació entre les varibles numèriques.
pairs(~ age + creatinine_phosphokinase + ejection_fraction + platelets +
        serum_creatinine + serum_sodium + time, data = heart_data,
      lower.panel = lower.panel,               #Configuració del panell inferior
      upper.panel = upper.panel)             #Configuració del panell superior
```



Per un moment es podria pensar que les dades relacionades amb la sang hauríen de tenir certa relació amb l'edat i que per tant els valors d'aquestes variables hauríen d'estar relacionats. Amb aquest idea a la ment el que farem serà dur a terme un test de correlació entre la dimensió `age` i la resta, així com entre la dimensió `time` i la resta i finalment entre elles.

La hipòtesi nul · la en un "correlation test" afirma que la correlació té un valor de 0, o sigui que NO existeix relació entre les 2 variables avaluades. Per tant si:

- El p-value és inferior a un nivell de significància de $\alpha = 0.05$ refusarem la hipòtesi nul · la.
- En cas contrari NO la refusarem.

Ara farem 2 grups de variables numèriques i confirmarem si realment existeix o no correlació entre les dimensions del 1º grup i les del 2º grup:

- Les que no aporten dades d'elements sanguinis, `age` i `time`.
- Les que si aporten dades, `creatinine_phosphokinase`, `ejection_fraction`, `platelets`, `serum_creatinine`, `serum_sodium`

```
cor.Age.Creat <- cor.test(formula = ~ age + creatinine_phosphokinase, data = heart_data,
                          method="spearman", exact = FALSE)
cor.Age.Eject <- cor.test(heart_data$age,heart_data$ejection_fraction,
                          method="spearman", exact = FALSE)
cor.Age.Plat <- cor.test(heart_data$age,heart_data$platelets
                         ,method="spearman", exact = FALSE)
cor.Age.SerumCreat <- cor.test(heart_data$age,heart_data$serum_creatinine,
                               method="spearman", exact = FALSE)
cor.Age.SerumSod <- cor.test(heart_data$age,heart_data$serum_sodium,
```

```
                              method="spearman", exact = FALSE)
cor.Age.Time <- cor.test(heart_data$age,heart_data$time,
                         method="spearman", exact = FALSE)
cor.Time.Creat <- cor.test(formula = ~ time + creatinine_phosphokinase, data = heart_data,
                           method="spearman", exact = FALSE)
cor.Time.Eject <- cor.test(heart_data$time,heart_data$ejection_fraction,
                           method="spearman", exact = FALSE)
cor.Time.Plat <- cor.test(heart_data$time,heart_data$platelets,
                          method="spearman", exact = FALSE)
cor.Time.SerumCreat <- cor.test(heart_data$time,heart_data$serum_creatinine,
                                method="spearman", exact = FALSE)
cor.Time.SerumSod <- cor.test(heart_data$time,heart_data$serum_sodium,
                              method="spearman", exact = FALSE)

print(paste("Test Correlacio Age ~ Creatinine_Phosphokinase: ", cor.Age.Creat$p.value))
```

```
## [1] "Test Correlacio Age ~ Creatinine_Phosphokinase:  0.108225630146979"
```

```
print(paste("Test Correlacio Age ~ Ejection_Fraction: ", cor.Age.Eject$p.value))
```

```
## [1] "Test Correlacio Age ~ Ejection_Fraction:  0.201678292062916"
```

```
print(paste("Test Correlacio Age ~ Platelets: ", cor.Age.Plat$p.value))
```

```
## [1] "Test Correlacio Age ~ Platelets:  0.369414708566015"
```

```
print(paste("Test Correlacio Age ~ Serum_Creatinine: ", cor.Age.SerumCreat$p.value))
```

```
## [1] "Test Correlacio Age ~ Serum_Creatinine:  2.05756421694447e-06"
```

```
print(paste("Test Correlacio Age ~ Serum_Sodium: ", cor.Age.SerumSod$p.value))
```

```
## [1] "Test Correlacio Age ~ Serum_Sodium:  0.0790847470298784"
```

```
print(paste("Test Correlacio Age ~ Time: ", cor.Age.Time$p.value))
```

```
## [1] "Test Correlacio Age ~ Time:  0.000592299821090548"
```

```
print(paste("Test Correlacio Time ~ Creatinine_Phosphokinase: ", cor.Time.Creat$p.value))
```

```
## [1] "Test Correlacio Time ~ Creatinine_Phosphokinase:  0.0296139307700398"
```

```
print(paste("Test Correlacio Time ~ Ejection_Fraction: ", cor.Time.Eject$p.value))
```

```
## [1] "Test Correlacio Time ~ Ejection_Fraction:  0.223968597967032"
```

```
print(paste("Test Correlacio Time ~ Platelets: ", cor.Time.Plat$p.value))
```

```
## [1] "Test Correlacio Time ~ Platelets:  0.905050851276357"
```

```
print(paste("Test Correlacio Time ~ Serum_Creatinine: ", cor.Time.SerumCreat$p.value))
```

```
## [1] "Test Correlacio Time ~ Serum_Creatinine:  0.00526535058868687"
```

```
print(paste("Test Correlacio Time ~ Serum_Sodium: ", cor.Time.SerumSod$p.value))
```

```
## [1] "Test Correlacio Time ~ Serum_Sodium:  0.136116031864808"
```

Després d'aplicar el test podem observar com l'idea inicial que teníem no anava mal enfocada i si que existeixen algunes variables que tenen una petita correlació entre elles:

- `age` versus `serum_creatinine` amb una molt lleu correlació positiva.
- `age` versus `time` amb una molt lleu correlació negativa.
- `time` versus `creatinine_phosphokinase` amb una molt lleu correlació positiva.
- `time` versus `serum_creatinine` amb una molt lleu correlació negativa.

El fet de haver detectat que existeixen correlacions entre diferents variables ens permetrà poder utilitzar la regressió lineal entre aquestes.

A continuació mostrem una taula de freqüències absolutes on observem la quantitat d'observacions de cada dimensió que pertanyen a la classe "Si" o "No" de la dimensió `death_event`. D'entrada una dada que ens fa preguntar-nos alguna qüestió és que la quantitat d'objectes que tenen un valor "No" a la dimensió `smoking` és la mateixa que la quantitar de pacients que finalment no moren i el mateix passa amb els que tenen un valor "Si". Aquest fet no vol dir que els objectes siguin els mateixos, per veure aquesta dada haurem de fer una taula de freqüencies relatives entre les 2 variables que es podrà apreciar a continuació.

```
#Taula de frequències per la dimensió [Death_Event]
t1 <- table(heart_data$DEATH_EVENT)
t1 <- as.matrix(t1)

colnames(t1)[1] <- "Class"
t1 <- cbind(t1,Dim_Anaemia=c(table(heart_data$anaemia)[1],table(heart_data$anaemia)[2]))
t1 <- cbind(t1,Dim_Diabetes=c(table(heart_data$diabetes)[1],table(heart_data$diabetes)[2]))
t1 <- cbind(t1,Dim_Pressure=c(table(heart_data$high_blood_pressure)[1],
                              table(heart_data$high_blood_pressure)[2]))
t1 <- cbind(t1,Dim_Sex=c(table(heart_data$sex)[1],table(heart_data$sex)[2]))
t1 <- cbind(t1,Dim_Smoking=c(table(heart_data$smoking)[1],table(heart_data$smoking)[2]))
addmargins(t1)
```

```
##     Class Dim_Anaemia Dim_Diabetes Dim_Pressure Dim_Sex Dim_Smoking  Sum
## No    203         170          174          194     105         203 1049
## Si     96         129          125          105     194          96  745
## Sum   299         299          299          299     299         299 1794
```

Ara podem veure la quantitat d'objectes agrupats en funció del valor dels seus atributs, a les columnes hi trobarem les dades de l'última dimensió en aquest cas `smoking`. A les files hi trobarem les dimensions indicades a la funció d'esquerra a dreta sense tenir en compte l'última.
Per exemple si ens fixem en el valor:

- [**26**] sabem que Si fumava, NO tenia la pressió alta, NO tenia diabetes, NO tenia anemia, NO va morir.
- [**5**] sabem que SI fumava, SI tenia la pressió alta, SI tenia diabets, NO tenia anemia i SI va morir

```
ftable(heart_data$DEATH_EVENT,heart_data$anaemia,heart_data$diabetes,
       heart_data$high_blood_pressure,heart_data$smoking)
```

```
##                No Si
##
## No No No No    22 26
##          Si    16  6
##       Si No    27  8
##          Si    11  4
##    Si No No    18 12
##          Si    14  4
##       Si No    20  4
```

```
##          Si   9  2
## Si No No No  13  6
##          Si   6  3
##       Si No   7  4
##          Si   6  5
##    Si No No  12  3
##          Si   7  6
##       Si No   9  3
##          Si   6  0
```

Taula de freqüències relatives i absolutes entre les dimensions `death_event` i `aenemia`

```
#Taula de frequències absolutes deathEvent - anaemia
t1 <- table(heart_data$DEATH_EVENT,heart_data$anaemia)
addmargins(t1)
```

```
##
##        No  Si Sum
##   No  120  83 203
##   Si   50  46  96
##   Sum 170 129 299
```

```
#Taula de frequències relatives sobre el total de la mostra
prop.table(t1, margin=NULL) * 100
```

```
##
##            No       Si
##   No 40.13378 27.75920
##   Si 16.72241 15.38462
```

Taula de freqüències relatives i absolutes entre les dimensions `death_event` i `diabetes`

```
#Taula frequències absolutes [death_Event] - [diabetes]
t1 <- table(heart_data$DEATH_EVENT,heart_data$diabetes)
addmargins(t1)
```

```
##
##        No  Si Sum
##   No  118  85 203
##   Si   56  40  96
##   Sum 174 125 299
```

```
#Taula de frequències relatives sobre el total de la mostra.
prop.table(t1, margin=NULL) * 100
```

```
##
##            No       Si
##   No 39.46488 28.42809
##   Si 18.72910 13.37793
```

Taula de freqüències relatives i absolutes entre les dimensions `death_event` i `high_blood_pressure`

```
#Taula frequències absolutes [death_Event] - [high_bood_pressure]
t1 <- table(heart_data$DEATH_EVENT,heart_data$high_blood_pressure)
addmargins(t1)
```

```
##
##        No  Si Sum
##   No  137  66 203
```

```
##   Si   57  39  96
##   Sum 194 105 299
```

```
#Taula de frequències relatives sobre el total de la mostra
prop.table(t1, margin=NULL) * 100
```

```
##
##            No       Si
##   No 45.81940 22.07358
##   Si 19.06355 13.04348
```

Taula de freqüències relatives i absolutes entre les dimensions `death_event` i `sex`

```
#Taula frequències absolutes [death_Event] - [sex]
t1 <- table(heart_data$DEATH_EVENT,heart_data$sex)
addmargins(t1)
```

```
##
##       femeni masculi Sum
##   No      71     132 203
##   Si      34      62  96
##   Sum    105     194 299
```

```
#Taula de frequències relatives sobre el total de la mostra
prop.table(t1, margin=NULL) * 100
```

```
##
##        femeni  masculi
##   No 23.74582 44.14716
##   Si 11.37124 20.73579
```

Taula de freqüències relatives i absolutes entre les dimensions `death_event` i `smoking`

```
#Taula frequències absolutes [death_Event] - [smoking]
t1 <- table(heart_data$DEATH_EVENT,heart_data$smoking)
addmargins(t1)
```

```
##
##        No  Si Sum
##   No  137  66 203
##   Si   66  30  96
##   Sum 203  96 299
```

```
#Taula de frequències relatives sobre el total de la mostra
prop.table(t1, margin=NULL) * 100
```

```
##
##            No       Si
##   No 45.81940 22.07358
##   Si 22.07358 10.03344
```

Un cop ja hem creat taules de contingències entre cadascuna de les variables i la variable *class* ja estem preparats per esbrinar si aquestes són o no independents respecte de la variable *class*. Per aconseguir-ho utilitzarem el test Chi-Quadrat.

L'hipòtesi nul · la d'aquest test és que les 2 variables són independents

```
#Test Chi-Squared entre cadascuna de les variables i la variable class
chi1 <- chisq.test(heart_data$DEATH_EVENT,heart_data$anaemia)
chi2 <- chisq.test(heart_data$DEATH_EVENT,heart_data$diabetes)
```

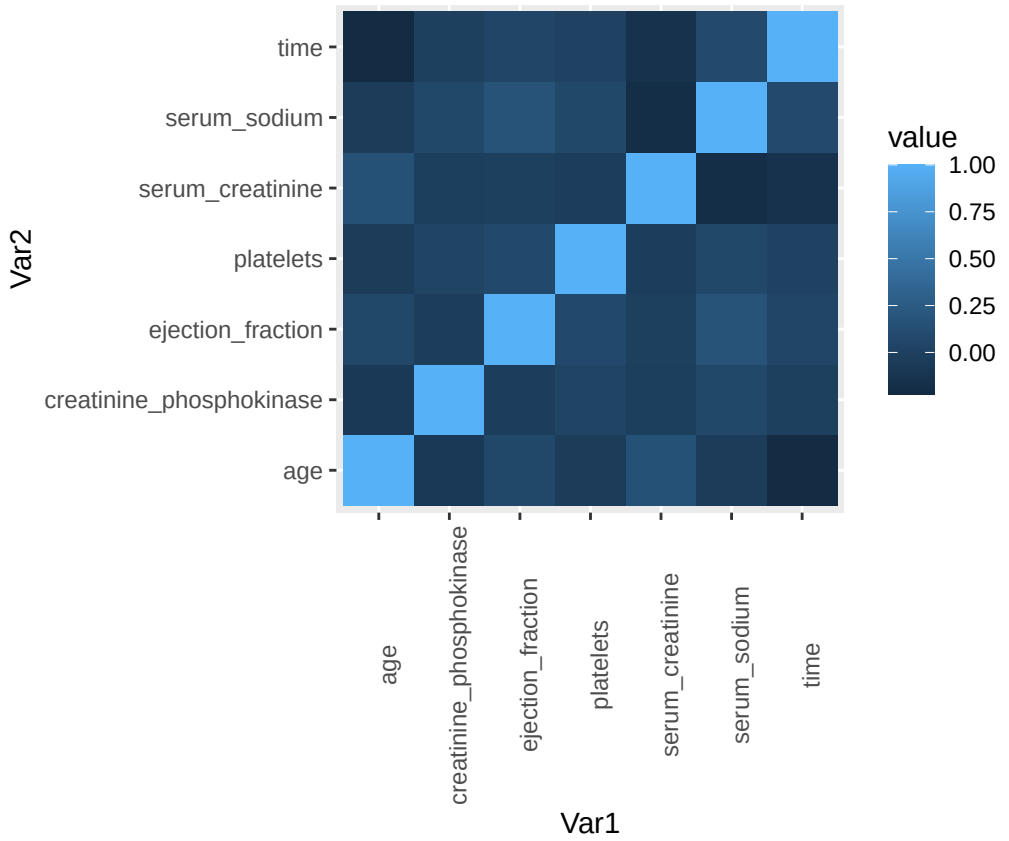
```r
chi3 <- chisq.test(heart_data$DEATH_EVENT,heart_data$high_blood_pressure)
chi4 <- chisq.test(heart_data$DEATH_EVENT,heart_data$smoking)
t <- c(chi1$p.value, chi2$p.value, chi3$p.value, chi4$p.value)
#Es crea una matriu de 4 files x 1 columna
matriXsquared <- matrix(t,nrow=4,ncol=1)
#Donem nom a la columna
colnames(matriXsquared)[1] <- "P-Value"
#Afegim la columna "Xsquared" a la matriu prèviament creada
matriXsquared <- cbind(matriXsquared,Xsquared=c(chi1$statistic, chi2$statistic,
                                                chi2$statistic, chi2$statistic))
#Donem nom a les diferents files
rownames(matriXsquared) <- c("Death_Event - Anaemia","Death_Event - Diabetes",
                             "Death_Event - Pressure","Death_Event - Smoking")
matriXsquared
```

```
##                           P-Value     Xsquared
## Death_Event - Anaemia  0.3073161 1.042175e+00
## Death_Event - Diabetes 1.0000000 2.161684e-30
## Death_Event - Pressure 0.2141034 2.161684e-30
## Death_Event - Smoking  0.9317653 2.161684e-30
```

Seguint el principi de refutació indicat prèviament i observant aquesta taula podem dir que tots els valors *p-value* mostrats són més grans que el nivell de significança i que per tant es compleix la hipòtesi nul · la o sigui aquestes variables respecte a la dimensió de la classe són independents.

A continuació utilitzarem un altre mètode per veure la correlació de les variables, ho farem mitjançant una anàlisi multivariant i observarem si obtenim els mateixos resultats que amb l'anàlisi anterior.

```r
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()
```

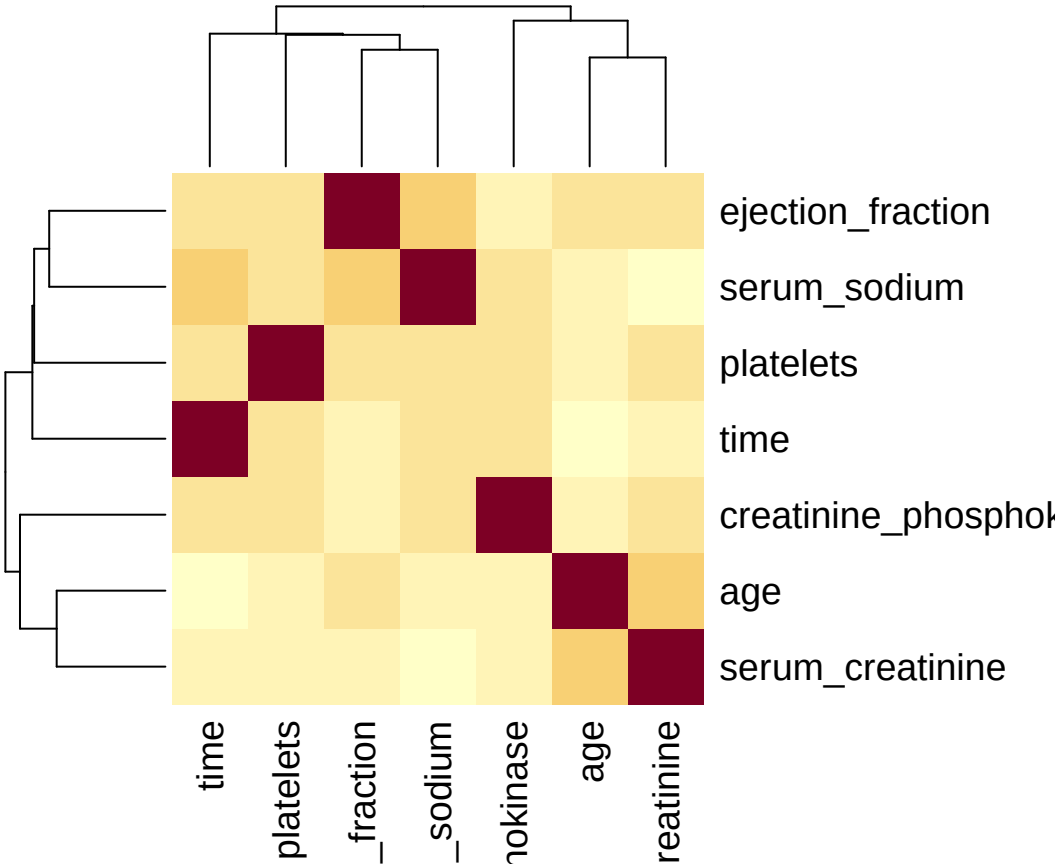


Com més intentistat de color, vol dir que més correlacionades estan les variables. Observem amb l'anàlisi multivariant, les variables que tenen una correlació més alta són:

- 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*.

4.3.2 Contrast hipòtesis

La segona prova que realitzarem serà un contrast d'hipòtesi sobre 2 mostres per poder determinar si el nivell de `creatinine_phosphokinase` és major amb els pacients que acaben morint o no. Aprofitarem i també ho farem amb la dimensió `platelets`.

Com la distribució de les dades d'aquestes dimensions no segueixen una Normal utilitzarem el test de *Mann–Whitney–Wilcoxon (WMW)*, també conegut com *Wilcoxon rank-sum test* o *u-test*. Referent en aquest test hem de donar constància d'alguns fets:

- Tenim poques observacions a les mostres.
- No tenim un total coneixement de les poblacions d'on s'han extret aquestes dades.

A continuació definirem les hipòtesis nul · la i alternativa:

Hipòtesis nul · la $(H_0) : \mu_1 = \mu_2$

Hipòtesis alternativa $(H_1) : \mu_1 \neq \mu_2$

Aquest és un test no paràmetric que contrasta si dues mostres procedeixen de 2 poblacions equidistribuides i que necesita d'uns mínims per a donar un resultat vàlid:

- Les dades han de ser independents.
- Les dades s'han de poder ordenar de més gran a més petit.

- Les dades de les mostres no han de seguir necessàriament una Normal.
- Les mostres han de complir amb el principi de Homocedasticitat i per comprovar-ho utilitzarem el test de *Fligner-Killen*.

Comparació entre les mostres `Death_Event="Si" ~ Creatinine_Phosphokinase` i `Death_Event="No" ~ Creatinine_Phosphoskinase`:

```
heart_data.SiDeath.CreatPhos <- heart_data[DEATH_EVENT == "Si",]$creatinine_phosphokinase
heart_data.NoDeath.CreatPhos <- heart_data[DEATH_EVENT == "No",]$creatinine_phosphokinase
#Ttest Fligner-Killen per validar si les variances de les 2 mostres són iguals o no
fligner.test(x = list(heart_data.SiDeath.CreatPhos,heart_data.NoDeath.CreatPhos))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(heart_data.SiDeath.CreatPhos, heart_data.NoDeath.CreatPhos)
## Fligner-Killeen:med chi-squared = 0.26845, df = 1, p-value = 0.6044
```

```
#Test "Mann-Whitney-Wilcoxon.
resp1 <- wilcox.test(heart_data.SiDeath.CreatPhos, heart_data.NoDeath.CreatPhos,
                     mu = 0, paired = FALSE, conf.int = 0.95, alernative="great")
resp1
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  heart_data.SiDeath.CreatPhos and heart_data.NoDeath.CreatPhos
## W = 10028, p-value = 0.684
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -32.99996  52.00005
## sample estimates:
## difference in location
##              0.9999748
```

- El test *Fligner-Killen* ha retornat un p-value = 0.6044, o sigui major que el nivell de significància 0.05 i per tant complim el principi indicat prèviament per que les 2 variances són homogenies.
- El test *Mann–Whitney–Wilcoxon* ha retornat un p-value = 0.6585, o sigui major que el nivell de significància 0.05 i per tant hi trobem suficients evidències per poder dir que els membres de la 1º mostra tenen una major probabilitat d'estar per sobre dels de la 2ª mostra.

Amb això en ment si que podem concloure que els pacients que finalment han mort tenien un nivell de `creatinine_phosphokinase` més alt que els que NO han mort.

Comparació entre les mostres `Death_Event="Si" ~ Platelets` i `Death_Event="No" ~ Platelets`

```
heart_data.SiDeath.SerumCreat <- heart_data[DEATH_EVENT == "Si", ]$platelets
heart_data.NoDeath.SerumCreat <- heart_data[DEATH_EVENT == "No", ]$platelets
fligner.test(x = list(heart_data.SiDeath.SerumCreat,heart_data.NoDeath.SerumCreat))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(heart_data.SiDeath.SerumCreat, heart_data.NoDeath.SerumCreat)
## Fligner-Killeen:med chi-squared = 2.3222, df = 1, p-value = 0.1275
```

```r
resp2 <- wilcox.test(heart_data.SiDeath.SerumCreat, heart_data.NoDeath.SerumCreat,
                     mu = 0, paired = FALSE, conf.int = 0.95, alternative = "great")
resp2
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  heart_data.SiDeath.SerumCreat and heart_data.NoDeath.SerumCreat
## W = 9187.5, p-value = 0.7876
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -24000    Inf
## sample estimates:
## difference in location
##                  -7000
```

- El test *Fligner-Killen* ha retornat un p-value = 0.1275, major que el nivell de significança 0.05 i per tant complim el principi indicat prèviament perquè les 2 variances són homogenies.

- El test *Mann–Whitney–Wilcoxon* ha retornat un p-value = 0.7876, o sigui major que el nivell de significànça 0.05 i per tant hi trobem suficients evidències per poder dir que els membres de la primera mostra (μ_1) tenen una major probabilitat d'estar per sobre dels de la segona mostra (μ_2).

Podem concloure que els pacients que finalment han mort tenien un nivell de `platelets` més alt que els que NO han mort.

4.3.3 Regressió

Quan tenim un model on s'avalua la influència de tot un conjunt de variables sobre una variable resposta i la variables és dicotòmica, ens trobem davant d'un model de regressió logística.

Observant el nostre joc de dades veiem clarament que compleix aquests requisits i per tant passarem a crear el model.

Com hem pogut veure en analitzar les variables numèriques no totes estan en la mateixa unitat de mesura, els diferents rangs que hi trobem a les dimensions són:

- `age` -> [40,95]
- `creatinine_phosphokinase` -> [23,7861]
- `ejection_fraction` -> [14,80]
- `platelets` -> [25100,850000]
- `serum_creatinine` -> [0.5,9.4]
- `serum_sodium` -> [113,148]
- `time` -> [4,285]

Per tant a l'hora d'obtenir quina proporció del resultat final aporta cada variable necessitarem estandaritzar-les.

Per aconseguir-ho utilitzarem *"Min-Max Normalization"*, els càlculs de la qual es troben dins de la funció *"normalize()"*. Un cop apliquem aquesta funció sobre les observacions d'una variable aconseguirem que totes les observacions pertanyin al rang [0,1].

```r
normalize <- function(x) {
  return((x- min(x)) / (max(x)-min(x)))
}
heart_data[,"age"] <- as.numeric(normalize(heart_data[,"age"]))
heart_data[,"creatinine_phosphokinase"] <- as.numeric(normalize(
  heart_data[,"creatinine_phosphokinase"]))
heart_data[,"ejection_fraction"] <- as.numeric(
  normalize(heart_data[,"ejection_fraction"]))
heart_data[,"platelets"] <- as.numeric(normalize(heart_data[,"platelets"]))
heart_data[,"serum_creatinine"] <- as.numeric(
  normalize(heart_data[,"serum_creatinine"]))
heart_data[,"serum_sodium"] <- as.numeric(normalize(heart_data[,"serum_sodium"]))
heart_data[,"time"] <- as.numeric(normalize(heart_data[,"time"]))
```

Fem un model només amb les variables categòriques:

```r
if (!require(ISLR)) {install.packages("ISLR")}

#Model variables categòriques.
glm.fitToM1 <- glm(DEATH_EVENT ~ anaemia + diabetes + smoking + high_blood_pressure,
                   data = heart_data, family = binomial)
#Obtenim el valor estimat, l'error standard, el z-score i el p-value dels coeficients.
summary(glm.fitToM1)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ anaemia + diabetes + smoking + high_blood_pressure,
##     family = binomial, data = heart_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0219  -0.8912  -0.7941   1.4633   1.6208
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.991830   0.251935  -3.937 8.26e-05 ***
## anaemiaSi              0.273396   0.251605   1.087    0.277
## diabetesSi            -0.000761   0.255318  -0.003    0.998
## smokingSi             -0.007851   0.272540  -0.029    0.977
## high_blood_pressureSi  0.340941   0.257249   1.325    0.185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 372.28  on 294  degrees of freedom
## AIC: 382.28
##
## Number of Fisher Scoring iterations: 4
```

```r
#Obtenim els predictors
glm.probsToM1 <- predict(glm.fitToM1,type = "response")
#Després de varies proves observem que el valor del predictor més gran és 0.4067315
glm.probsToM1[ which(glm.probsToM1[] > 0.4067)  ]
```

```
##        14        15        19        21        27        49        51        52
## 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315
##        84        90        96       124       144       155       161       215
## 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315
##       218       228       236       244       255
## 0.4067315 0.4067315 0.4067315 0.4067315 0.4067315
```

Un cop arribats aquí veiem que la probabilitat màxima que ens ofereixen els predictors d'aquest model és de 40.67%.

Ara fem un model amb les variables numèriques:

```
#Fem un model només amb les variables numèriques.
glm.fitToM2 <- glm(DEATH_EVENT ~ age + creatinine_phosphokinase + ejection_fraction +
                     platelets + serum_creatinine + serum_sodium + time,
                   data = heart_data, family = binomial)
#Obtenim el valor estimat, l'error standard, el z-score i el p-value dels coeficients.
summary(glm.fitToM2)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + creatinine_phosphokinase +
##     ejection_fraction + platelets + serum_creatinine + serum_sodium +
##     time, family = binomial, data = heart_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1976  -0.5716  -0.2281   0.4545   2.7638
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 3.1887     1.1816   2.699 0.006962 **
## age                         2.4160     0.8411   2.872 0.004075 **
## creatinine_phosphokinase    1.4471     1.3300   1.088 0.276565
## ejection_fraction          -4.7865     1.0469  -4.572 4.83e-06 ***
## platelets                  -0.7458     1.5094  -0.494 0.621238
## serum_creatinine            6.0829     1.5831   3.842 0.000122 ***
## serum_sodium               -2.2654     1.3535  -1.674 0.094179 .
## time                       -5.8725     0.8249  -7.119 1.09e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 221.93  on 291  degrees of freedom
## AIC: 237.93
##
## Number of Fisher Scoring iterations: 6
```

```
#Obtenim els predictors
glm.probsToM2 <- predict(glm.fitToM2,type = "response")
#Obtenim els predictors que ens donen una probabilitat per sobre del 98%
glm.probsToM2[ which(glm.probsToM2[] > 0.98)  ]
```

```
##         1         5        10        29        49
## 0.9826037 0.9920770 0.9995806 0.9808007 0.9939552
```

Un cop arribats aquí veiem que la probabilitat màxima que ens ofereixen els predictors d'aquest model és de 99.95%.

Ara fem un model amb totes les variables (numèriques i categòriques):

```
attach(heart_data)
#Fem un model amb totes les variables
glm.fitToM3 <- glm(DEATH_EVENT ~ age + creatinine_phosphokinase + ejection_fraction +
                     platelets + serum_creatinine + serum_sodium + time + anaemia +
                     diabetes + smoking + high_blood_pressure, data = heart_data,
                   family = binomial)
#Obtenim el valor estimat, l'error standard, el z-score i el p-value dels coeficients.
summary(glm.fitToM3)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + creatinine_phosphokinase +
##     ejection_fraction + platelets + serum_creatinine + serum_sodium +
##     time + anaemia + diabetes + smoking + high_blood_pressure,
##     family = binomial, data = heart_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2664  -0.5711  -0.2253   0.4655   2.7771
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.14719    1.25636   2.505 0.012245 *
## age                        2.49938    0.85428   2.926 0.003437 **
## creatinine_phosphokinase   1.51478    1.37699   1.100 0.271304
## ejection_fraction         -4.87093    1.05829  -4.603 4.17e-06 ***
## platelets                 -0.73835    1.52302  -0.485 0.627823
## serum_creatinine           6.04085    1.61514   3.740 0.000184 ***
## serum_sodium              -2.21956    1.37342  -1.616 0.106076
## time                      -5.86979    0.84058  -6.983 2.89e-12 ***
## anaemiaSi                  0.03020    0.35695   0.085 0.932576
## diabetesSi                 0.17885    0.34970   0.511 0.609053
## smokingSi                 -0.21998    0.37476  -0.587 0.557224
## high_blood_pressureSi     -0.04155    0.35388  -0.117 0.906529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 221.24  on 287  degrees of freedom
## AIC: 245.24
##
## Number of Fisher Scoring iterations: 6
```

```
#Obtenim els predictors
glm.probsToM3 <- predict(glm.fitToM3,type = "response")
#Obtenim els predictors que ens donen una probabilitat per sobre del 98%
glm.probsToM3[ which(glm.probsToM3[] > 0.98)  ]
```

```
##         1         5        10        29        49
```

```
## 0.9823431 0.9934726 0.9994611 0.9805993 0.9940372
```

Un cop arribats aquí veiem que la probabilitat màxima que ens ofereixen els predictors d'aquest model és de 99.94%.

Un cop vist que els models 2 i 3 tenen pràcticament la mateixa eficiència, desestimarem el model 1 que només utilitzava variables categòriques com a variables independents.

Ara passarem a fer una predicció per veure si el pacient acaba morint o No en funció de les variables independents utilitzades.

Es crea un vector de valors "Si" i "No" en funció de si les probabilitats dels predictors del vector [predToM2] estan o no per sobre del valor 99'95%.

Es crea una taula de contingencia entre el nou vector creat i la dimensió de classe del nostre dataset per saber quantes coincidències hi trobem, hem de pensar que l'éxit es troba en les dades que es troben a la diagonal.

Finalment calculem la "mean" i observem que el nostre model ens ofereix un 68'22% de fiabilitat.

```
glm.predToM2 <- ifelse(glm.probsToM2 > 0.9995, "Si", "No")
addmargins(table(glm.predToM2,heart_data$DEATH_EVENT))
```

```
##
## glm.predToM2  No  Si Sum
##          No  203  95 298
##          Si    0   1   1
##          Sum 203  96 299
```

```
mean(glm.predToM2 == heart_data$DEATH_EVENT)
```

```
## [1] 0.6822742
```

Apliquem el mateix que hem dit prèviament i obtenim una fiabilitat del 68'22%.

```
glm.predToM3 <- ifelse(glm.probsToM3 > 0.9994, "Si", "No")
addmargins(table(glm.predToM3,heart_data$DEATH_EVENT))
```

```
##
## glm.predToM3  No  Si Sum
##          No  203  95 298
##          Si    0   1   1
##          Sum 203  96 299
```

```
mean(glm.predToM3 == heart_data$DEATH_EVENT)
```

```
## [1] 0.6822742
```

```
newData1 <- data.frame(age = 0.63636364, anaemia = "Si",
                       creatinine_phosphokinase = 0.071319214,
                       diabetes = "Si", ejection_fraction = 0.09090909,
                       high_blood_pressure = "Si", platelets = 0.29082313,
                       serum_creatinine = 0.15730337, serum_sodium = 0.48571429,
                       sex = "femeni", smoking = "Si", time = 0.000000000)
predict(glm.fitToM3, newData1)
```

```
##        1
## 4.007885
```

```
glm.fitToM5 <- glm(DEATH_EVENT ~ age + creatinine_phosphokinase + ejection_fraction +
                     platelets + serum_creatinine + serum_sodium + time + anaemia +
                     diabetes + smoking + high_blood_pressure,
```

```
                  data = heart_data, family = binomial)
glm.probsToM5 <- predict(glm.fitToM5, newData1, type = "response")
glm.probsToM5[ which(glm.probsToM5[] > 0.98)  ]
```

```
##         1
## 0.9821525
```

```
glm.predToM5 <- ifelse(glm.probsToM5 > 0.98, "Si", "No")
addmargins(table(glm.predToM5,heart_data[15,"DEATH_EVENT"]))
```

```
##
## glm.predToM5 No Si Sum
##          Si   1  0   1
##          Sum  1  0   1
```

```
mean(glm.predToM5 == heart_data[15,"DEATH_EVENT"])
```

```
## [1] 0
```

Hem pogut apreciar que el model que utilitza només variables numèriques ofereix la mateixa fiabilitat que el model que utilitza totes les variables i per tant per fer la predicció d'un nou objecte o sigui d'un nou pacient utiltizarem el model3 així farem ús de totes les variables independents.

Per validar el comportament del model3 ens inventarem les dades d'un pacient i farem que el model realitzi una predicció, que serà comparada posteriorment amb un valor "No" de la dimensió class. Com podem apreciar a la taula de contingencia la predicció ens dona un "Si", pero com ho hem comparat amb un valor "No" la proporció d'encert ha estat 0. Hem de dir a més que hem fet la prova amb dades de pacients que es troben dins del dataframe i el resultat predit ha estat el correcte.

5 Resolució del problema.

Al llarg d'aquest treball s'han realitzat diferents tractaments de les dades amb l'objectiu d'aconseguir respondre algunes preguntes i a més poder obtenir un model que sigui capaç de fer prediccions amb una alta fiabilitat.

Hem començat analitzant les dades per observar els seus tipus i si hi havia *missing values* i *outliers*. Pel que fa als primers hem vist que aquest joc de dades no en tenia cap, però respecte als segons sí que n'hem trobat. El fet de no tenir *missing values* ens ha donat confiança respecte a la possibilitat de NO tenir dades amb errades en el joc de dades i hem decidit acceptar aquestes dades tal com estan.

Hem pogut apreciar l'existència o NO de correlacions entre les diferents variables, tant numèriques com categòriques, per saber com unes podien o no tenir influència en les altres. Quant a les categòriques s'ha demostrat que no existia cap relació entre elles respecte a la dimensió de la classe i sobre les numèriques hem pogut apreciar alguna petita relació entre unes poques variables.

El test d'hipòtesi ens ha permès saber la influència d'algunes variables, com `creatinine_phosphokinase` i `platelets` sobre el fet que el pacient acabi morint o NO. En ambdós casos hem vist com realment els valors d'aquests elements sanguinis eren majors en els pacients que finalment morien.

Finalment cal dir que per a poder realitzar alguns dels objectius citats anteriorment hem hagut de sotmetre a les diferents variables a proves de Normalitat i d'Homocedasticitat, per poder aconseguir el màxim de confiança en els resultats generats.

6 Referències

R MarkDown. https://bookdown.org/yihui/rmarkdown/

Correlation Test. http://www.sthda.com/english/wiki/correlation-test-between-two-variables-in-r

AprendeR. https://aprender-uib.github.io/AprendeR2/

Homoscedasticitat. https://www.cienciadedatos.net/documentos/9_homogeneidad_de_varianza_homocedasticidad.html

Handbook R. https://rcompanion.org/handbook/index.html

The Pirates Guide of R. https://bookdown.org/ndphillips/YaRrr/

Test Significance Correlation Coefficient. https://courses.lumenlearning.com/introstats1/chapter/testing-the-significance-of-the-correlation-coefficient/

Overview Categorical - Continuous. https://medium.com/@outside2SDs/an-overview-of-correlation-measures-between-categorical-and-continuous-variables-4c7f85610365

Contingency Tables. https://www.datacamp.com/community/tutorials/contingency-tables-r

Contingency analysis https://www.datacamp.com/community/tutorials/contingency-analysis-r

Analisis Correlació. https://www.maximaformacion.es/blog-dat/analisis-de-correlacion-en-r/

Chi-Square. http://www.sthda.com/english/wiki/chi-square-test-of-independence-in-r

Normalization. https://www.datanovia.com/en/blog/how-to-normalize-and-standardize-data-in-r-for-great-heatmap-visualization/

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.”